PROGRAMMABLE LOGIC DEVICE WITH METHOD OF PRREVENTING READBACK

Publication number: WO0244875 (A2)

Publication date: 2002-06-06

Inventor(s): PANG RAYMOND C; SZE WALTER N; THENDEAN JOHN M; TRIMBERGER STEPHEN M; WONG JENNIFER

Applicant(s): XILINX INC [US]

Classification:

- international: ***H03K19/173; G06F17/50; G06F21/00; G09C1/00; H04L9/10;*** **H03K19/173; G06F17/50; G06F21/00; G09C1/00; H04L9/10;** (IPC1-7): G06F1/00

- European: G06F17/50D4; G06F21/00N1C5

Application number: WO2001US45055 20011128

Priority number(s): US20000724975 20001128

Also published as:

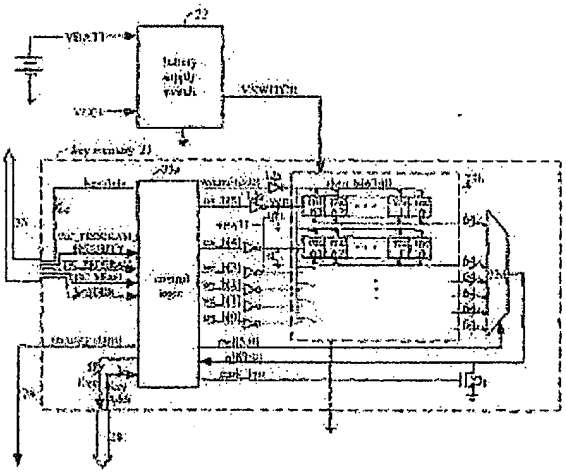
WO0244875 (A3)
US6981153 (B1)
JP2004515180 (T)
EP1358530 (A2)
CA2429597 (A1)

Cited documents:

US5970142 (A)
US2001015919 (A1)
WO0049717 (A2)

Abstract of **WO 0244875 (A2)**

It is sometimes desirable to protect a design used in a PLD from being copied. If the design is stored in a diferent device from the PLD and read into the PLD through a bitstream, the design may be encrypted as it is read into the PLD and decrypted within the PLD before being loaded into configuration memory cells for configuring the PLD. According to the invention, in such a device, a method is provided to prevent the design from being read back from the PLD in its decrypted state if it had been encrypted when loaded into the PLD.



(19)日本国特許庁(JP)　(12)公表特許公報(A)　(11)特許出願公表番号
特表2004-515180
(P2004-515180A)
(43)公表日　平成16年5月20日(2004.5.20)

(51)Int.Cl.[7]		F I			テーマコード（参考）
H03K	19/173	H03K	19/173	101	5J042
G09C	1/00	G09C	1/00	650Z	5J104
H04L	9/10	H04L	9/00	621A	

審査請求 未請求　予備審査請求 有　（全 87 頁）

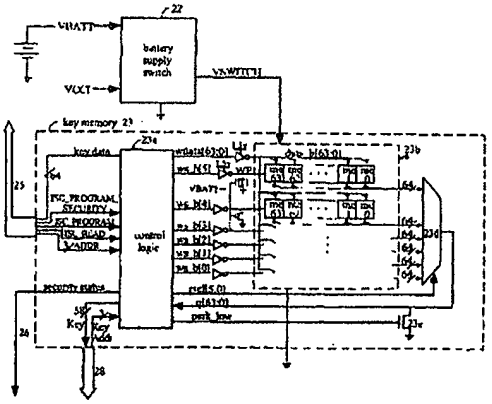
(21)出願番号　特願2002-546975 (P2002-546975)
(86)(22)出願日　平成13年11月28日 (2001.11.28)
(85)翻訳文提出日　平成15年5月27日 (2003.5.27)
(86)国際出願番号　PCT/US2001/045055
(87)国際公開番号　WO2002/044875
(87)国際公開日　平成14年6月6日 (2002.6.6)
(31)優先権主張番号　09/724,975
(32)優先日　平成12年11月28日 (2000.11.28)
(33)優先権主張国　米国 (US)
(81)指定国　EP (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), CA, JP

(71)出願人　591025439
ザイリンクス　インコーポレイテッド
ＸＩＬＩＮＸ　ＩＮＣＯＲＰＯＲＡＴＥＤ
アメリカ合衆国　カリフォルニア州　９５１２４－３４００　サン　ホセ　ロジック　ドライブ　２１００
(74)代理人　100064746
弁理士　深見　久郎
(74)代理人　100085132
弁理士　森田　俊雄
(74)代理人　100083703
弁理士　仲村　義平
(74)代理人　100096781
弁理士　堀井　豊

最終頁に続く

(54)【発明の名称】リードバックを防止する方法を備えたプログラマブルロジックデバイス

(57)【要約】
ＰＬＤ内で使用される設計をコピーされないように保護することがしばしば望まれている。設計がＰＬＤとは異なるデバイスに記憶され、ビットストリームを通じてＰＬＤ内に読込まれる場合、設計はＰＬＤ内に読込まれる際に暗号化され、ＰＬＤ内で復号されてから構成メモリセル内にロードされてＰＬＤを構成することがある。この発明に従うと、このようなデバイスにおいて、設計がＰＬＤ内へのロードの際に暗号化された場合、設計がその復号された状態でＰＬＤからリードバックされることを防止するための方法が提供される。



【特許請求の範囲】
【請求項１】
ＰＬＤを構成するための設計を保護するための方法であって、
ＰＬＤ内に、設計を表現する暗号化されたビットストリームをロードするステップと、
前記ＰＬＤ内のビットストリームを復号して、前記設計を表現する暗号化されていないビットストリームを生成するステップと、
暗号化されていない前記設計で前記ＰＬＤを構成するステップと、
前記暗号化されていないビットストリームのリードバックを禁止するステップとを含む、方法。
【請求項２】
ＰＬＤを構成するための設計を保護する方法であって、
ＰＬＤ内に、設計を表現する暗号化されたビットストリームをロードするステップと、
前記ＰＬＤ内のビットストリームを復号して、前記設計を表現する暗号化されていないビットストリームを生成するステップと、
前記暗号化されていないビットストリームで前記ＰＬＤを構成するステップと、
前記ＰＬＤが部分的に再構成されることを禁止するステップとを含む、方法。
【請求項３】
前記ＰＬＤが部分的に再構成されることを禁止する前記ステップはさらに、ユーザの設計の或る部分が前記ＰＬＤの別の部分に再配置されることを防ぐステップを含む、請求項２に記載のＰＬＤ。
【発明の詳細な説明】
【０００１】
【発明の分野】
この発明はＰＬＤに関し、より特定的に、ビットストリームを介してＰＬＤ内にロードされる設計の保護に関する。
【０００２】
【発明の背景】
ＰＬＤ（プログラマブルロジックデバイス）とは、設計者が選択したデジタル論理関数を実行する集積回路構造である。ＰＬＤはロジックブロックと相互接続線とを含み、典型的にロジックブロックおよび相互接続はともにプログラム可能である。１つの一般的な種類のＰＬＤがＦＰＧＡ（フィールドプログラマブルロジックデバイス）であり、ここでロジックブロックは典型的にルックアップ表およびフリップフロップを含み、典型的にその入力信号のあらゆる関数を生成および記憶できる。別の種類はＣＰＬＤ（コンプレックスプログラマブルロジックデバイス）であり、ここではロジックブロックはＡＮＤ関数、およびＯＲ関数を実行し、入力信号の選択はプログラム可能である。
【０００３】
ＰＬＤの外部でのビットストリームの記憶に関する問題
ＰＬＤ内で実現される設計は複雑化しており、ＰＬＤ内で実現すべき設計を完成させデバッグするにはしばしば数ヵ月を要する。ＰＬＤが一部をなすシステム内に設計が入れられ、利益のため販売される際、設計者にとって誰か他の者がこの設計努力をコピーするという結果は望ましいものではない。設計者はしばしば設計を企業秘密にしておこうと望む。多くのＰＬＤ、特にＦＰＧＡは揮発性構成メモリを使用し、これにはＰＬＤに電源が投入されるたびにＰＲＯＭなど外部のデバイスからロードされることになる。構成データはＰＬＤの外部に記憶され、構成アクセスポートを通じて伝送されることになるため、攻撃者、すなわち構成アクセスポート上のデータをたとえばボードトレース上にプローブを置くことにより監視する者によって、設計の機密は容易に侵される恐れがある。
【０００４】
現在の解決策およびその欠点
設計を暗号化する試みがなされてきているが、設計を攻撃者に対して安全にし、かつ正規のユーザによる使用を容易にすることは困難である。暗号化アルゴリズムは問題ではない

。データブロックを暗号化するためには、いくつかの暗号化アルゴリズム、たとえば標準的なデータ暗号規格（ＤＥＳ）およびより安全な高度暗号規格（ＡＥＳ）のアルゴリズムなどが公知である。暗号ブロック連鎖（ＣＢＣ）のプロセスでは、暗号化されていないデータワードと次の暗号化されたデータワードとのＸＯＲをとってから復号が行なわれ、これによってＤＥＳまたはＡＥＳはデータのシリアルストリームを暗号化できるので、ＰＬＤを構成するためのビットストリームを暗号化するのに好適である。設計を暗号化するのに用いる鍵は何らかの形で、ＰＬＤと、設計を復号する構造との間で安全なやり方で通信されることになり、こうして設計がＰＬＤにより復号され、ＰＬＤを構成するのに用いられ得る。次に、暗号化されていない設計を用いてＰＬＤが一旦構成されると、設計を無認可で発見されないよう保護し続けなければならない。

【０００５】

「構成に関する問題：電源投入、揮発性、安全性、電池のバックアップ（Ｃｏｎｆｉｇｕｒａｔｉｏｎ Ｉｓｓｕｅｓ：Ｐｏｗｅｒ－ｕｐ，Ｖｏｌａｔｉｌｉｔｙ，Ｓｅｃｕｒｉｔｙ，Ｂａｔｔｅｒｙ Ｂａｃｋ－ｕｐ）」と題された、ザイリンクス・インコーポレイテッド（Ｘｉｌｉｎｘ，Ｉｎｃ．）のピーター・アルフキー（Ｐｅｔｅｒ Ａｌｆｋｅ）による１９９７年１１月２４日の刊行物には、設計を保護するために、ＦＰＧＡ内に特定のアーキテクチャ上の特徴を有さない、既存のＦＰＧＡデバイスでの設計を保護するためにとられ得る、いくつかのステップが記載されている。１つの方法は、ＦＰＧＡ内に設計構成データをロードしてから構成データのソースを外すが、電池を用いてＦＰＧＡに対して連続的に電力を維持し、一方でＦＰＧＡを待機非動作モードに保つことである。しかし電池に対する電力の要件のため、この方法は大規模なＦＰＧＡデバイスにとって非実用的なものとなっている。

【０００６】

別の可能性は不揮発性構成メモリである。デバイスの販売前に設計が工場でロードされる場合、構成されたＰＬＤデバイスの購入者にとって、設計がどのようなものであるかを判断することは困難である。しかし逆行分析によって、プログラムされたデバイスの覆いを取去り、金属層を除去し、不揮発性メモリセルに化学処理を施すと、どのメモリセルが充電されたかが明らかとなる可能性があり、こうして攻撃者は設計を知ることができる恐れがある。さらに不揮発性メモリには、標準的なＣＭＯＳプロセス技術よりも複雑で経費のかかるプロセス技術が必要であり、市場に出るまでにより多くの時間がかかってしまう。

【０００７】

ＰＬＤの不揮発性メモリ内に復号鍵を記憶させ、暗号化されたビットストリームをＰＬＤ内にロードし、ＰＬＤ内の鍵を用いてビットストリームを復号することもまた公知である。こうすれば、攻撃者がビットストリームをＰＬＤ内へのロード中に読出すことが防止され、かつ、電力がＰＬＤから外されても鍵が維持される。このような機構は、オースティン（Ａｕｓｔｉｎ）による米国特許第５，３８８，１５７号に記載されている。しかしこの構造はユーザの設計をあらゆる態様の攻撃から保護するわけではない。

【０００８】

設計保護に加え、ユーザによってはデータ保護もまた必要である。ＰＬＤが電力を失っても失われてはならないデータを、ユーザがＰＬＤ内に生成することがある。このようなデータは保護されることが望ましい。

【０００９】

したがって、便利で信頼性が高く、かつ安全な設計保護方法に対する必要性がなお存在している。

【００１０】

【発明の概要】

この発明は、ＰＬＤを無認可の使用およびデータの紛失から保護するためのいくつかの構造および方法を提供する。

【００１１】

電源投入の際にロードされなければならない静的ＲＡＭメモリによってＰＬＤが構成され

る場合、構成データはデバイスへのロードの際に保護されなければならない。先行技術においてこれは、構成データを暗号化して集積回路デバイスの外部にあるメモリに記憶させ、１つ以上の復号鍵をＰＬＤ内にロードして電源切断時に鍵をＰＬＤ内に維持し、鍵を用いて構成データを復号する復号回路をＰＬＤ内に含め、復号された構成データをＰＬＤ内で生成し、復号された構成データを用いてＰＬＤを構成することによって達成される。

【００１２】

さらなる安全性のために、不揮発性メモリを用いて鍵を保存する代わりに、この発明は好ましくはＰＬＤに接続された電池を用いて、ＰＬＤから電力が外される際に鍵を保存する。不揮発性メモリ内に鍵を記憶させたＰＬＤを取外し、ＰＬＤの覆いを外し、不揮発性ビットのうちどれが論理１にプログラムされどれが論理０にプログラムされているかを観察することは可能であるが、静的メモリセルにのみ記憶された鍵の内容を判断することは極めて困難であり、なぜなら、鍵を記憶させるためだけでも鍵を記憶するメモリセルには電力が維持されなければならず、連続的にＰＬＤに対し動作電力がある間にＰＬＤの覆いを外し、層を取除き、プローブをかけなければならないであろうからである。

【００１３】

一旦ＰＬＤ内にロードされた設計を攻撃者が盗み得るやり方

鍵が十分な安全性を提供していなければ、攻撃者は暗号化コードを破って鍵の値を判断する可能性がある。周知のデータ暗号規格ＤＥＳが用いていた暗号鍵は５６ビットのものであり、鍵解読用の高性能のコンピュータによって数時間で破られてしまった。ＤＥＳはブルース・シュナイヤー（Ｂｒｕｃｅ Ｓｃｈｎｅｉｅｒ）によって、ブルース・シュナイヤー著、ジョン・ワイリー＆サンズ・インコーポレイテッド（Ｊｏｈｎ Ｗｉｌｅｙ ＆ Ｓｏｎｓ， Ｉｎｃ．）刊行の「応用暗号技術第２版：プロトコル、アルゴリズム、およびＣにおけるソースコード（Ａｐｐｌｉｅｄ Ｃｒｙｐｔｏｇｒａｐｈｙ Ｓｅｃｏｎｄ Ｅｄｉｔｉｏｎ：ｐｒｏｔｏｃｏｌｓ，ａｌｇｏｒｉｔｈｍｓ，ａｎｄ ｓｏｕｒｃｅ ｃｏｄｅ ｉｎ Ｃ）」著作権１９９６年の２６５～２７８頁で説明されている。このような周知の暗号規格を用いることが所望であれば、安全性を増大させるために構成データは数度、各回につき異なった鍵を用いて暗号化され、こうして暗号化を繰返すたびに約2^{56}だけ暗号化コードを強化することがある。または構成データを、一番目の鍵を用いて暗号化し、２番目の鍵を用いて復号し、３番目の鍵を用いて暗号化することがあり、この組合せは三重ＤＥＳ規格の一部である。その他の暗号化アルゴリズムを用いることもできるが、安全性は鍵に存するためアルゴリズムを秘密にしておく必要はない。暗号化の方法が対称であれば、暗号化に用いた同じ鍵がＰＬＤ内に記憶され、復号では逆の順序で用いられる。

【００１４】

多数の鍵を提供するＰＬＤでは、用いるべき鍵の数とあらゆる鍵のアドレスとが、暗号化されていないビットストリームで提供される場合、攻撃者は１回につき１つずつ鍵に攻撃をかけて鍵の値をより容易に判断できる可能性がある。このような攻撃を回避するため、いくつの鍵を用いるのか、および、鍵がセット内の最後の鍵か、またはさらなる鍵が続くのかについての指示を、ビットストリーム内でなく鍵内に記憶することによって追加の安全性を達成する。

【００１５】

ビットストリームがＰＬＤ内にロードされた後にビットストリームをリードバックする選択肢をＰＬＤが提供する場合、攻撃者が用い得る別の方法はこのビットストリームをリードバックすることである。設計を攻撃するこの方法を回避するため、一実施例で、リードバックを提供しかつ暗号化も提供するＰＬＤは、暗号化が使用された場合にはリードバックのフィーチャを禁止する能力を含む。別の実施例では、リードバックの能力を提供するＰＬＤは構成データをそのリードバック前に暗号化する。

【００１６】

加えていくつかのＰＬＤは、部分的構成（いくつかの構成アドレスを特定して設計におけるいくつかの部分をロードする）および部分的再構成（既存の設計が消去されずに新たな

設計データがロードされる）の選択肢を提供する。ＰＬＤがこれらの選択肢を提供する場合、攻撃者はＰＬＤを部分的に再構成し、連続する設計部分を可視にして、おそらくは設計全体を知る可能性がある。このような攻撃を回避するため、一実施例では、暗号化された設計がロードされたＰＬＤの部分的な構成および再構成は却下される。別の実施例では、いくつかの構成アドレスが特定され得るが、アドレスは暗号化される。

【００１７】

さらに別の攻撃態様は、ＰＬＤの安全性状態を示すビットを反転させようとする試みである。動作電圧を低下または上昇させる、温度を変化させる、および或るポートに雑音を加えることが考えられる。このようなビット反転から保護するため、安全にされたビットストリームでＰＬＤが動作しているときには安全モードのフラグをセットし、一実施例では、このフラグが非セットされるとすべての構成データが消去される。デバイスがまだ動作している間に再構成を許さない別の実施例では、構成データはビットストリームのいずれかが送信される前に消去される。

【００１８】

別の攻撃態様は、暗号化されたビットストリームの或る部分を再配置することによって、暗号化されていないときにこれら部分が、設計者の意図していないＰＬＤの可視部分に置かれるようにすることである。この再配置を防ぐため、暗号化および復号のプロセスでアドレス情報を使用することによって、設計者が意図したＰＬＤ場所とは異なるＰＬＤ場所に暗号化されたビットストリームの一部を送ると、意味のないデータへの異なる復号が行なわれるようにする。暗号ブロック連鎖（ＣＢＣ）は、この結果を達成する１つの有効な手段である。暗号ブロック連鎖では、復号されたデータパケット（ブロック）が次のデータブロックとＸＯＲ関数を用いて組合されてから次のブロックが復号され、こうして各データブロックについての暗号化されたデータは、これに先行するあらゆるブロックと、これらブロックの順序とに依存する。同一のデータブロックは、これに先行するデータブロックの値に依存して異なった値へ暗号化される。こうして、ブロックの順序が変えられるとビットストリームは正しく復号しないが、それは暗号化されたビットストリームが再配列されるところがその後のデータをスクランブルするからである。さらに、初期ＣＢＣ値に変更を加えてデータのアドレスを組込ませ、復号されたデータが、正しく復号するための特定の場所に置かれるよう強制することができる。

【００１９】

これに代えて、設計の一部が暗号化されかつ別の部分が暗号化されないことをＰＬＤが許す場合、攻撃者は暗号化されていない部分を暗号化された部分に加えることができ、これは設計の暗号化された部分についての情報を読出すことになる。したがって、設計すべてが暗号化されているか、または設計がいずれも暗号化されていないかのいずれかを可能にし、混合をなくすことによって、追加の安全性が達成される。これに加えて一実施例では、データ暗号化の際に、構成データについての単一の開始アドレスに続く単一のフルチップ構成のみを許すことによって、追加の安全性がもたらされる。

【００２０】

さらに、試験およびデバッグを手頃にし、かつＰＬＤ製造業者とその顧客（すなわちＰＬＤを構成するための設計を生産する設計者）との自由な通信を可能にするために、ＰＬＤは暗号化される動作モードと暗号化されない動作モードとの両方を有し、暗号化されるモードで動作する際、ＰＬＤ内への構成データのロードを制御する構成ビットストリームの部分はまだ暗号化されていない。

【００２１】

別の攻撃態様として、ＰＬＤ製造業者が、構成データをロードするためのアドレスおよびヘッダ情報を含む構成ビットストリームフォーマットについて自由に情報を提供、および使用される暗号化方法についての情報を提供する場合、この周知の情報を暗号化すると、暗号鍵があり得る発見に対して暴露されてしまうことになるだろう。このような暴露は、実際の構成データのみを暗号化し、制御情報を暗号化されないままにすることによって回避される。

【００２２】
ＰＬＤ製造業者が安全モードおよび非安全モードの両方で鍵メモリを用いることを許す場合、攻撃者は単に鍵メモリを非安全モードに置いて鍵を読出すことによって鍵を知ることができるであろう。このような攻撃を回避するため、ＰＬＤ製造業者は、回路であって、鍵メモリが非安全モードへ動かされた際、すべての鍵と、ＰＬＤ内にロードされたあらゆる構成データとの消去を引起こす回路を含める。
【００２３】
［詳細な説明］
図１は先行技術のＦＰＧＡ１０の構造を示す。ＦＰＧＡはプログラマブルロジック１１を含み、これは典型的に（１）ルックアップ表の組合せ論理関数生成器を伴ったロジックブロックと、ルックアップ表出力およびその他の値を記憶するためのフリップフロップと、プログラマブルロジックの論理能力を増大させるための論理ゲートおよびマルチプレクサとを含み、さらに（２）ＦＰＧＡのまわりに信号を経路付けするための経路付け線およびプログラム可能相互接続点と、（３）ＦＰＧＡの外部ピンおよび経路付け線間で信号を駆動するための入出力ブロックとを含む。
【００２４】
ＦＰＧＡはさらに構成メモリ１２を含み、これは経路付けトランジスタをオンにし、マルチプレクサを制御し、ルックアップ表を記憶し、さらに入出力ブロックを制御するが、これらすべてはＦＰＧＡを構成して開発者の所望の関数を実行することを目的としたものである。バス１６は構成メモリ１２をプログラマブルロジック１１に接続し、典型的にＦＰＧＡ全体にわたって位置付けられた分散した１組の制御線である。いくつかのザイリンクス製品（たとえばＸＣ６２００）はバス１７を含んでおり、これによってプログラマブルロジック１１は、構成ロジック１４がプログラミング情報を構成メモリ１２に送ることを引起こす。このような構造はキーン（Ｋｅａｎ）による米国特許第５，７０５，９３８号で説明されている。
【００２５】
ＦＰＧＡ１０はさらに、ＪＴＡＧポート２０とインターフェイスするためのＪＴＡＧロジックブロック１３を含み、これは特に、ＦＰＧＡが置かれることになるボードの試験を意図したものである。ＪＴＡＧロジックブロック１３はＩＥＥＥ規格１５３２を満たすが、これはＩＥＥＥ規格１１４９．１の上位セットである。ＪＴＡＧはボードレベルでの設計のデバッグを可能にする。
【００２６】
最後に、ＦＰＧＡ１０は構成ロジック１４を含み、これは構成アクセスポート２１上の、外部ソース１５からの構成ビットストリームに応答し、さらにＪＴＡＧロジックブロック１３にインターフェイスする。構成アクセスポート２１上のビットストリームはワード、一実施例では３２ビットワードとして扱われる。通常はビットストリームの初めまたはその近くにあるワードのいくつかは構成プロセスを準備するために用いられ、たとえば構成メモリフレームの長さ、および構成データについての開始アドレスを含む。バス１９は構成ロジック１４とＪＴＡＧロジックブロック１３との間の通信を可能にし、こうしてＪＴＡＧポートはもう１つの構成アクセスポートして使用可能である。バス１８は構成ロジックブロック１４と構成メモリ１２との間の通信を可能にする。具体的には、メモリ１２内で構成フレームを選択するためのアドレスと、読み書き動作を行なうための制御信号と、構成メモリ１２内にロードまたは構成メモリ１２からリードバックするためのデータとを伝える。
【００２７】
構成ロジックブロック１４は命令およびデータを受取り、命令に従ってデータを処理する。これら命令はビットストリームとして構成ロジック１４に入る。命令、またはヘッダには通常、処理を受けることになるデータが続く。図２ａは例示的なビットストリーム構造を示す。ヘッダＡはアクションを特定し、単一のワード、すなわちデータＡが続くことを特定する。ヘッダＢはアクションを特定し、この場合は４ワードのデータが続いて処理を

受けることを特定する。

【００２８】

図２ｂは、ザイリンクス・インコーポレイテッドから入手可能なバーテックス（Ｒ）（Ｖｉｒｔｅｘ）デバイスで用いられるビットストリーム内の３２ビットヘッダワードについてのデフォルトフォーマット（フォーマットタイプ００１）を示す（バーテックスは、この発明の譲受人であるザイリンクス・インコーポレイテッドの登録商標である）。このフォーマットは、フォーマットタイプ（００１）を示すための３つのビットと、オペコードを特定するための２つのビットと、構成ロジックレジスタアドレスのための１６ビットと、ワードカウントのための１１ビットとを含む。オペコードは読出動作、書込動作または動作なしを指定し得る。たとえば００は動作なしを指定し、０１は読出を指定し、１０は書込を指定することがあり得る。ワードカウントのための１１ビットは2^{11}ワードまたは２０４８ワードを特定し得る。図２ｃに示すように、ワードカウントがこれよりも大きければ、フォーマットタイプ００１におけるワードカウントビットは０００００００００００に設定され、フォーマットタイプ００１のヘッダにはフォーマットタイプ２のヘッダが続く。フォーマットタイプ２はワードカウントを特定するために２７ビットを使用し、こうして2^{27}ワードまたは２６８万ワードを特定し得る。

【００２９】

図２ｄは、バーテックスのビットストリームについてのヘッダにより構成ロジック１４のレジスタ内にロードされ得るような制御情報を示す。たとえば、構成ロジックレジスタアドレス００００を有する（フォーマット００１の）ヘッダは、次の３２ビットデータワードが巡回冗長検査（ＣＲＣ）レジスタ内にロードされるべきであることを特定する。（バーテックスのデバイスは１６ビット巡回冗長検査値を用いるため、いくつかのビットは０で埋められる。）ヘッダがアドレス０００１を含む場合、次のデータはフレームアドレスレジスタ内にロードされて構成メモリ１２内でフレーム（列）を特定し、こうしてデータが受取られるまたは供給されることになる。

【００３０】

図２ｂに示す構成ロジックレジスタアドレス（１６ビット）は、図２ｄの左の列に示す４ビットの値を与え、これら値は、次の３２ビットデータワードを置くべき構成ロジック１４（図１）内のレジスタの１つを選択する。フレーム長レジスタ（アドレス１０１１）は、構成データをロードすることになるフレームの長さを特定する。（フレーム長または列の高さはＰＬＤのサイズに依存する。通常、より大きいＰＬＤはより高い列またはより長いフレームを有する。ＰＬＤ内に異なる構造を設けてデータワードをフレーム内に置く代わりに、ビットストリーム内でフレーム長を特定しレジスタ内でフレーム長を記憶することにより、異なったサイズのＰＬＤについて内部構成ロジックが同一となることができる。）

【００３１】

リードバックのためには、読出コマンドがオペコードフィールド内に置かれ、フレームデータ出力レジスタがアドレス指定され、これに（必要であればコマンドヘッダフォーマット２を用いた）ワードカウントが続く。特定された数のワードは、フレームアドレスレジスタ内で特定されたアドレスから始まって構成メモリ１２からリードバックされ、構成アクセスポート２１またはＪＴＡＧポート２０へシフトされる（リードバックデータは、リードバック命令を発行したポートへ戻される）。

【００３２】

ビットストリームヘッダまたはヘッダ対（図２ｂおよび図２ｃ）でワードカウントが特定されるとカウンタがセットされ、これはデータワードがロードされるのに伴ってカウントダウンする。多くの構成ロジックレジスタアドレスではワードカウントは１である。しかし、構成データがロードまたはリードバックされていることを示すための００１０または００１１の構成ロジックアドレスをビットストリームヘッダが有する場合、ワードカウントははるかに大きくなるであろう。これは図２ｃのヘッダフォーマット２を使用する場合である。フレームデータ入力レジスタ（アドレス００１０）を通じて構成メモリ１２内に

ロードされたデータ、またはフレームデータ出力レジスタ（アドレス００１１）を通じてリードアウトされたデータは設計データと呼ばれるが、それはこのデータがＦＰＧＡに設計を実現させる、または設計の状態を示すからである。他のレジスタデータは制御データであるが、それはこれらデータによって、ロジックの構成またはリードバック中に構成ロジックがどう振舞うかが制御されるからである。

【００３３】

バーテックスのデバイスの構成についてのさらなる詳細は、ザイリンクス・インコーポレイテッド（この発明の譲受人）、９５１２４　カリフォルニア州、サン・ノゼ、ロジック・ドライブ（Ｌｏｇｉｃ　Ｄｒｉｖｅ，　Ｓａｎ　Ｊｏｓｅ，　ＣＡ）、２１００による、２０００年１０月９日刊行の「バーテックス構成ガイド（Ｖｉｒｔｅｘ　Ｃｏｎｆｉｇｕｒａｔｉｏｎ　Ｇｕｉｄｅ）」に記載されている。

【００３４】

構成ロジック１４は典型的に、入って来る構成ビットストリームに対して巡回冗長検査（エリクソン（Ｅｒｉｃｋｓｏｎ）による米国特許第５，３２１，７０４号を参照、または先に参照したバーテックス構成ガイドの３９頁から４０頁を参照）を行ない、構成されている部分のフレーム長と構成データのワードカウントとを示すヘッダビットを読出し、どこに構成データをロードすべきかを同定するアドレス命令を読出し、構成データのフレームを収集し、アドレスに示された構成メモリ１２の列にこれらフレームをロードする。構成ロジック１４はさらに構成メモリ１２から外部の場所への構成データおよびフリップフロップ値のリードバックを制御する。ザイリンクス・インコーポレイテッドから入手可能なバーテックスＦＰＧＡでは、ＪＴＡＧポート２０または構成アクセスポート２１を通じてリードバックが行なわれ得る。

【００３５】

構成ロジック１４はさらにプログラマブルロジック１１から構成データを受取り得る。ＦＰＧＡの或る部分がＦＰＧＡの別の部分を構成する先行技術のＦＰＧＡ構造についてのさらなる情報は、キーンによる米国特許第５，７０５，９３８号に記載されている。バーテックスのアーキテクチャに類似のＦＰＧＡアーキテクチャについてのさらなる情報は、ヤング（Ｙｏｕｎｇ）他による米国特許第５，９１４，６１６号に記載されている。この発明の譲受人であるザイリンクス・インコーポレイテッドから入手可能なバーテックス製品で用いられるビットストリームのフォーマットは、ザイリンクス・インコーポレイテッド、９５１２４　カリフォルニア州、サン・ノゼ、ロジックドライブ、２１００から入手可能である、２０００年１０月４日刊行の「バーテックスＦＰＧＡシリーズ構成およびリードバック（Ｖｉｒｔｅｘ　ＦＰＧＡ　Ｓｅｒｉｅｓ　Ｃｏｎｆｉｇｕｒａｔｉｏｎ　ａｎｄ　Ｒｅａｄｂａｃｋ）」と題された出願ノートＸＡＰＰ１３８に記載されている。

【００３６】

復号を伴うＰＬＤ

図３は、この発明の一実施例に従うＦＰＧＡ（ＰＬＤの１種）のブロック図を示す。いくつかの要素は図１に示すものと同じであり、これらには同じ参照番号を付し、もはや説明は行なわない。これに加えて図３は、拡張された構成ロジックユニット２９、復号器２４および鍵メモリ２３を含む。図３は、鍵メモリ２３がＪＴＡＧアクセスポート２０からバス２５によってロードされる実施例を示す。実施例によっては、鍵メモリ２３には別のポートを通じてロードされる。バス２５は、データ、アドレス、および読み書き動作を行なうための制御信号を伝え、ＪＴＡＧポート２０からの復号鍵のプログラミングを可能にする。一実施例では、バス２６が構成ポートからの鍵のプログラミングを可能にする。別の実施例ではバス２６はなくされる。さらに別の実施例では、バス２６はあり、バス２５はなくされる。ここでさらに説明する実施例では、バス２６は安全性に関するデータを鍵メモリ２３から構成ロジック２９に伝える。一実施例では、バス２７は暗号化された構成データを構成ロジック２９から復号器２４に伝え、復号された構成データを構成ロジック２９に戻す。バス２８は復号器２４がデータの復号のために鍵にアクセスすることを可能にする。暗号化されたデータが図３の構造にロードされる際、ロード中のビットストリーム

を監視する攻撃者は暗号化されたビットストリームのみを受取るため、この方法でユーザの設計を知ることはできない。

【００３７】

部分的に暗号化されたビットストリーム

この発明の別の局面に従うと、ビットストリームは２つの部分、すなわち暗号化され得るまたはされ得ないユーザの設計を表現するデータ部分と、ビットストリームのロードを制御する制御部分とを含む（たとえば連続するビットストリーム部分がロードされるべきＰＬＤ内の列アドレスを与え、ロード動作の信頼性を調べるための巡回冗長検査（ＣＲＣ）コードと、暗号ブロック連鎖（ＣＢＣ）のためのスタータ数とを提供するが、この技術は、暗号化されたデータの発生の頻度から復号データが推理され得る「辞書攻撃」を防止する）。この発明の好ましい実施例では、データ部分は暗号化され得るが、制御部分は暗号化されない。これによって追加の安全性がもたらされるが、それはＰＬＤ製造業者はビットストリームの制御フィーチャを自由に記述する必要があり、もしこの比較的よく知られた制御情報が暗号化されれば、攻撃者はこの情報を解読し、この情報を用いてビットストリーム全体を解読できるというおそれがあるからである。さらに、ビットストリームの制御部分を暗号化されないままにしておくことによって、ＰＬＤによる情報の使用がより容易となる。

【００３８】

構成データがロードされるアドレスの順序が攻撃者にとって設計の分析に有用となり得る場合に用いられる別の実施例では、構成データのアドレスもまた暗号化されるが、構成ビットストリームにある他の制御情報は暗号化されないままである。

【００３９】

ビットストリームフォーマット

図４ａ～４ｄは、図２ａ～２ｄに示す先行技術の製品の構成ロジック１４のレジスタおよびビットストリームフォーマットと比較した場合の、構成ロジック２９のレジスタおよびビットストリームフォーマットにおける違いを例示する。図４ａに示すように、ビットストリームはやはりヘッダワードを含み、これにデータワードが続く。典型的な構成では、いくつかの制御データワードは、暗号化された構成データが始まる前にレジスタ内にロードされる。図４ａは、３つのヘッダワード、すなわちヘッダＡ、ヘッダＢおよびヘッダＣに各々３つの暗号化されない制御データワード、すなわちデータＡ、データＢおよびデータＣが続く例を示す。（実際の構成では、３つを上回る制御データワードが与えられるであろう。）次にヘッダＤは、暗号化された構成データが続くであろうことを特定し、多数のワード、すなわち暗号化された構成データであるデータ１Ｄ、データ２Ｄ、データ３Ｄなどが続く。図４ａではこれらワードに影をつけ、このデータが暗号化されていることを強調する。

【００４０】

図４ｂおよび図４ｃに示すように、第４のオペコードが追加されている。動作なしについての値００、復号のない読み書きについての０１および１０に追加された新たな値１１は、書込が復号を伴うべきであることを特定する。復号を用いるべきであることを特定するのにどのコードまたはどの方法を用いるかは重要ではなく、またそれをオペコードで特定することすら重要ではない。重要なのは、任意の暗号化および復号が許されかつ示され、このため設計者がこの選択肢を利用できることのみである。図４ｄの実施例では２つの新たな構成ロジックレジスタが追加されている。アドレス１１００および１１０１では、暗号ブロック連鎖（ＣＢＣ）のスタータ値、および初期暗号鍵のためのアドレスを保持するためのレジスタが示される。

【００４１】

任意の暗号化

この発明の別の局面に従うと、ＰＬＤはビットストリームの暗号化されたデータ部分および暗号化されていないデータ部分の両方を受入れることができる。ビットストリームの制御部分は、ビットストリームのデータ部分が暗号化されているかどうかを示す。ビットス

トリームのデータ部分が暗号化されていれば、これはＰＬＤ内で復号器へ迂回させられ、復号の後にＰＬＤを構成する。データ部分が暗号化されていなければ、これは迂回させられず、ＰＬＤを構成するために直接用いられる。

【００４２】

ビットストリームを暗号化しないことが好ましい場合がある。設計のデバッグ中に用いられる或る試験作業では、構成情報をリードバックすることが必須である。構成上の問題の診断は、（特に暗号化が問題と関わりを有しているかどうかを設計者が判断しようとする場合）暗号化のステップがまだ行なわれていない方がより単純となる。また、数人の設計者がＰＬＤの複数部分で実現されるべきコードを書いており、ＰＬＤの異なった部分が異なったときに構成されるべきである場合、ビットストリームの部分すべてを可視にして、ＰＬＤを部分的に再構成可能にすることが必要であろう。

【００４３】

図５ａおよび図５ｂは、この発明の一実施例における暗号化されていないビットストリームと暗号化されたビットストリームとの違いを例示するために、まず暗号化されていない、次に暗号化された、同じ設計を表わす例示のビットストリーム部分を示す。実際のビットストリームは図の右に０および１を含み、左のテキストは含まない。左にあるテキストは右にあるビットの意味を説明するために入れたものである。これらビットストリーム部分は図４ｂ～４ｄで例示したコマンドを用いる。図５ａの暗号化されていないバージョンと、図５ｂの暗号化されたバージョンとの違いを強調するため、相違箇所を太字で示す。

【００４４】

図５ａを参照して、ダミーワード（すべて１として解釈される一定のハイの信号）および１と０との特定のパターンを有する同期ワード（ｓｙｎｃ　ｗｏｒｄ）の後、次のワードはタイプ００１のものであり、１０のオペコードを有し、０００００００００００１００００のアドレスと、０００００００００００１のワードカウントとを有する。したがってこのワードはコマンドレジスタＣＭＤをアドレス指定し、１つのワードがここに書込まれるであろうことを特定する。図５ａではビットストリームの左に注が付され、このワードがタイプ（Ｔｙｐｅ）１であり、１つのワード（ｗｏｒｄ）のＣＭＤへの書込を指示することが示される。続くワード１１１はコマンドレジスタＣＭＤに置かれるべきデータであり、ＣＲＣ（巡回冗長検査）レジスタをリセットする。（好ましい実施例でＰＬＤは、ビットストリームがロードされるのに伴いビットストリームからＣＲＣ値を算出するための、エリクソンによる米国特許第５，５９８，４２４号に記載のものなど図には示さない回路を含み、誤ったビットのロードを引起こしかねないビットストリーム電圧の不調から保護する。）次に、ヘッダワードによって、フォーマットはやはりタイプ１であり、フレーム長レジスタＦＬＲに１ワードを書込むべきことを特定することが特定される。続くデータワード１１００１はフレーム長（２５ワード）を特定する。同様に、いくつかの追加のヘッダおよびデータワードが続くが、ここにはフレームアドレスレジスタＦＡＲに書込むべきワードを特定するヘッダが含まれる。この場合、続くデータワードはデータがアドレス０で始まることを示す。最後に、これらレジスタがロードされた後、フレームデータ入力レジスタＦＤＲＩにデータを書込むためのコマンドが来るが、かなりのデータが書込まれることになるため、ワードカウントは０００００００００００００として与えられ、タイプ２のヘッダによって、１０５３０ワードがＦＤＲＩレジスタに書込まれることが特定される。これはＰＬＤの構成を引起こす実際の設計データである。したがってビットストリーム内の次の１０５３０ワードは設計データである。最後に、データが正しくロードされたことを確かにするために、構成データの出所であるデバイスが算出したＣＲＣ値がロードされ、ＰＬＤの算出したＣＲＣ値と比較される。追加のコマンドおよびデータがロードされ、こうして構成が完了したこと、およびＰＬＤを動作モードに動かすべきことを示す。

【００４５】

図５ｂは図５ａと類似するが、データおよび注が太字で示されているところのみが異なる。図５ｂではデータは暗号化され、追加のコマンドを用いて初期鍵アドレスを与え、２つのワード（６４ビット）をＣＢＣ（暗号化ブロック連鎖）レジスタに書込む。次に、タイ

プ１のヘッダはオペコード１１を含み、データがフレームデータ入力レジスタＦＤＲＩに書込まれる前に復号されるであろうことを示す。タイプ２ヘッダが続いてやはりオペコード１１を有し、１０５３０ワードが復号されデータ入力レジスタＦＤＲＩに書込まれるべきであるという命令を与える。次に１０５３０個の暗号化されたデータワードが続く。次に、（暗号化された）データが正しくロードされたことを確認するためのＣＲＣワードが続く。最後に、追加のコマンドおよびデータが送られ、すべてが正しければＰＬＤを動作モードに置く。

【００４６】

復号プロセス

図６は、一実施例で任意の復号がどのように達成されるかを示す。図６は、構成ロジック２９と、復号器２４内に入るバス２７，２８とを詳細に示す。バス２７は以下のものを含む。

・構成ロジック２９内のレジスタアドレス１１０１（図４ｄ）から取られる３ビットの初期復号鍵アドレス“Ｉｎｉｔ＿ｋｅｙ＿ａｄｄｒ”、

・６４ビットの、変更を加えられた暗号ブロック連鎖値“ｍｏｄＣＢＣ”。この値は、構成ロジック２９内のレジスタアドレス１１００（図４ｄ）から取られる６４ビットＣＢＣ値のより低いオーダのビットを、レジスタ０００１で特定される２２ビットフレームアドレス値と取替えることによってもたらされる。

・ビットストリームから取られる、暗号化されたデータをロードするための６４線“Ｅｎｃｒｙｐｔｅｄ＿ｄａｔａ”、

・復号器２４が生成する復号されたデータを構成ロジック２９に戻すための６４線“Ｄｅｃｒｙｐｔｅｄ＿ｄａｔａ”、

・データが“Ｅｎｃｒｙｐｔｅｄ＿ｄａｔａ”線上にあり、復号器２４が復号を始めることができることを復号器２４に告げる信号“Ｅｎｃ＿ｄａｔａ＿ｒｄｙ”用の線、

・６４ビットワードに対する復号が完了して“Ｄｅｃｒｙｐｔｅｄ＿ｄａｔａ”線上で利用可能であることを構成ロジック２９に告げる信号“Ｄｅｃ＿ｄａｔａ＿ｒｄｙ”用の線、および

・たとえば鍵が或るセットの最初、中間または最後にあるべきかを指定する鍵メモリ内のビットによって特定される通りに鍵が用いられていない場合、構成ロジック２９に構成を中断させてこれに従い状態レジスタをセットさせるために、復号器２４により用いられるＢａｄ＿ｋｅｙ＿ｓｅｔ線。図４ｄに示す実施例では、状態レジスタはアドレス０１１１にあり、ビットのうち１つに論理１を記憶することによってＢａｄ＿ｋｅｙ＿ｓｅｔエラーが示される。

【００４７】

バス２８は以下のものを含む。

・鍵アドレス用の３線。鍵アドレスは最初はバス２７から与えられるものであるが、新たな鍵が用いられるたびに更新される。

【００４８】

・復号鍵用の５６線、および

・復号鍵が最初、中間、最後、または用いる唯一の鍵であるかを示すための２線。

【００４９】

設計再配置の防止

暗号化されたビットストリーム内の設計に対する１つの考えられる攻撃は、暗号化されたビットストリーム内でフレームアドレスレジスタ（開始アドレス）を変えることによって、これが復号されると、ＦＰＧＡの使用中には可視であるＦＰＧＡ部分へロードされるようにすることである。設計によってはブロックＲＡＭの内容は可視である。入出力ポートの構成はあらゆる設計で可視であり、このため構成ビットは判断可能である。したがって、連続する設計部分がＦＰＧＡの可視部分へ動かされれば、ＦＰＧＡが正しく機能しなかったとしても、攻撃者は再配置を繰返すことで、暗号化されていないビットストリームの内容を知ることができるであろう。

【００５０】
設計再配置を防止するため、一実施例では、ＤＥＳ規格で用いられる暗号ブロック連鎖方法で使用される初期値に対して変更を加える。図７ａおよび図７ｂはそれぞれこの発明に従い変更を加えられる三重ＤＥＳアルゴリズムの暗号化部分および復号部分を示す。標準的な暗号ブロック連鎖方法は、開始数（これは設計者が与えるものであっても、またはランダムに生成されるものであってもよい）と、暗号化すべきデータの１番目のワードとのＸＯＲを取って暗号化プロセスを始める。この発明に従い、ランダムな数の一部はアドレス情報、すなわちこの例では、データが構成メモリ１２にロードされることになる１番目のフレームの２２ビットアドレスと取替えられる。６４ビットの数であるスタータＣＢＣ値におけるｘのラベルを付した最下位ビットは、ｙのラベルを付したフレームアドレスと取替えられ、こうして変更を加えられた６４ビット値がもたらされるが、これはデータがロードされるアドレスに依存する。この変更を加えられたＣＢＣ値と、構成情報の１番目のワードであるワード１とのＸＯＲが取られる。次に暗号化アルゴリズムを用いて１番目の暗号化されたワードである暗号化ワード１をもたらし、これはビットストリーム内に置かれる。図７ａは外部暗号ブロック連鎖を伴った三重暗号化アルゴリズムを示し、これは１番目の鍵を用いた暗号化ステップenc_1を含み、これに２番目の鍵を用いた復号ステップdec_2が続き、これに３番目の鍵を用いた暗号化ステップenc_3が続く。この１番目の暗号化されたワードである暗号化ワード１と、２番目の、暗号化されていないワードであるワード２とのＸＯＲが取られ、暗号化プロセスを繰返して暗号化されたワード２をもたらす。すべての構成データが暗号化されるまでＸＯＲ連鎖を継続する。

【００５１】
図７ｂに示すように、ＰＬＤは逆のプロセスを行なって、復号されたワードを導くことになる。上記の暗号化の順番の場合、復号の順番は鍵３を用いた復号ステップdec_1、そして鍵２を用いた暗号化ステップenc_2、そして鍵１を用いた復号ステップdec_3となるであろう。重要なことに、復号ワード１を生成するための初期値の一部は暗号化および復号の両方に同じフレームアドレスを用いることになる。ビットストリームでなくＰＬＤが、フレームアドレスレジスタに記憶されたフレームアドレスから、変更を加えられたＣＢＣ値を生成し、これはまた、構成データをロードすべき構成メモリ１２のフレームを特定するためにも用いられる。したがって攻撃者が、データをロードすべきフレームアドレスを変えた場合、変更を加えられたＣＢＣ値はこれに従って変わり、構成データは正しく復号されない。

【００５２】
ＸＯＲステップは、暗号化の前に設計者のビットストリーム内にあった元のデータを生成する。たとえば元のデータ１＝復号ワード１となる。この復号された構成データはバス２７（図３）で構成ロジック２９へ送られる。

【００５３】
構成ロジック２９
構成ロジック２９は、任意の暗号化をサポートするための構造と、設計再配置および単一の鍵攻撃を防ぐための構造とを含む。図６に示すように、構成ロジック２９は保持レジスタ２９２、制御ロジック２９１、構成レジスタ（ＦＤＲＩ、ＦＡＲ、ＣＲＣ、およびｉｎｉｔＣＢＣを図示する）、復号器２４とのインターフェイス用マルチプレクサ２９４，２９５、６４ビットのアセンブリレジスタ２９７、およびレジスタ２９８，２９９（構成アクセスポート２１とのインターフェイス用）を含む。６４ビットシフトレジスタ２９９はデータを構成アクセスポート２１から受取り、このポートは１ビット幅のデータ用の単一のピンであっても、または８ビット幅のデータ用の８本のピンであってもよい。このデータは６４ビットのシフトレジスタ２９９内にロードされ、これはレジスタ２９９が一杯になるまで行なわれる。次に、好ましくはこれら６４ビットは６４ビット転送レジスタ２９８内に並列にシフトされる。ここからマルチプレクサ２９６ｂは右および左の３２ビットワードを交替で選択し、マルチプレクサ２９６ａは制御線Ｍに制御されるのに応じて、一回につきデータの３２ビットを、保持レジスタ２９２か、またはこれに代えてアセンブリ

レジスタ２９７のハイおよびローの部分のいずれかに送る。ビットストリームのロードが始まると、線Ｍおよび図示しないクロック信号は、マルチプレクサ２９６ａおよび２９６ｂが６４ビット転送レジスタ２９８から保持レジスタ２９２へデータを送ることを引起こす。ここからこれらワードは制御ロジック２９１に与えられる。ワードがヘッダであれば、制御ロジック２９１はワードを解釈する。続くデータが暗号されずに書込まれるべきであることをオペコードが示していれば、制御ロジック２９１はアドレスをバスＧに置いてレジスタを選択し、線Ｌ上に信号を置いてマルチプレクサ２９４がバスＢをバスＤに接続することを引起こし、続くワードをバスＢに与える。次のクロック信号（クロック信号は図示せず）の際、バスＤ上のデータはアドレス指定されたレジスタにロードされる。図４ｄに示すレジスタすべてはこのようにロードされ得る。初期暗号ブロック連鎖値をロードするためのｉｎｉｔＣＢＣレジスタは６４ビットレジスタであり、図５ｂに示し先に論じたように２つの連続する３２ビットワードを受取る。

【００５４】

（１）ｉｎｉｔＣＢＣレジスタに記憶された元のＣＢＣ値と、（２）ＦＡＲレジスタに記憶された初期フレームアドレスとから得られる、変更を加えられたＣＢＣ値は復号器２４に利用可能である。一実施例では、ＦＡＲレジスタにある初期フレームアドレスが使用するのはせいぜい３２ビットであるのに対し、ｉｎｉｔＣＢＣ値は６４ビットを使用する。図６の実施例では、変更を加えられたＣＢＣ値を与える６４ビットバスは、フレームアドレスレジスタＦＡＲからの２２ビットと、初期ＣＢＣレジスタからの４２ビットとを含む。この発明が提供する安全性にとっては重要なことであるが、この値は構成データをロードするところに依存することに注目されたい。もし攻撃者がＦＡＲレジスタの内容を変えることによって、暗号化されたデータを異なる場所にロードしようとすれば、復号器２４に送り込まれるｍｏｄＣＢＣ値もまた変わることになる。

【００５５】

或る数の構成データワードを復号するためのオペコードコマンドを制御ロジック２９１が受取ると、復号プロセスが始まる。制御線Ｍは、マルチプレクサ２９６ａがデータを転送レジスタ２９８から、アセンブリレジスタ２９７へと通じるバスＡに与えることを引起こす。制御バスＨは、暗号化されたデータのレジスタ２９７のハイ［３１：０］およびロー［３１：０］の部分にバスＡを交替で接続し、こうして復号すべき６４ビットワードが得られる。次に制御ロジック２９１はＥｎｃ＿ｄａｔａ＿ｒｄｙ信号をアサートし、この信号は、復号器２４がレジスタ２９７内のデータを復号することを引起こす。

【００５６】

復号器２４は復号を行なうために、鍵アドレスｋｅｙＡｄｄｒをバス２８によって鍵メモリ２３（図３）に与える。これは、鍵メモリ２３が、このアドレスにある５６ビット鍵を５６ビットＫｅｙ線上に返すことを引起こす。これはまた、鍵メモリ２３が、やはりこのアドレスで鍵データに記憶されている２つの追加のビット“Ｏｒｄｅｒ”を返すことを引起こす。１番目の復号鍵の場合、これら２つのビットはこれが１番目の鍵である、または唯一の鍵であることを示すことになる。もしそうでなければ、復号器２４はＢａｄ＿ｋｅｙ＿ｓｅｔ信号をアサートし、この信号は制御ロジック２９による構成動作の中断を引起こす。もしこれら２つのビットによって、鍵が１番目の鍵または唯一の鍵であることが示されている場合、復号器２４は、たとえば（シュナイヤーの同書に記載の）周知のＤＥＳアルゴリズムを用いて復号を行なう。鍵が唯一の鍵でなければ、次に復号器２４は鍵メモリ２３における次のアドレスで鍵を得て、これが中間または最後の鍵であることを２つのＯｒｄｅｒビットが示しているかどうかを調べる。もしそうでなければＢａｄ＿ｋｅｙ＿ｓｅｔ信号がアサートされ、構成は中断される。もしそうであれば復号が行なわれる。もしこれが中間の鍵であれば次の復号作業が行なわれる。もしこれが最後の鍵であれば、復号器２４は、復号されたワードと値ｍｏｄＣＢＣとのＸＯＲ関数をもたらす。次に復号器２４は結果として得られた値を６４ビットＤｅｃｒｙｐｔｅｄ＿ｄａｔａバスに置き、Ｄｅｃ＿ｄａｔａ＿ｒｄｙ信号をアサートする。この信号は制御ロジック２９１が信号を制御線Ｋに置くことを引起こし、こうしてマルチプレクサ２９５が６４ビットワードを２つ

のシーケンシャルな３２ビットワードに分割することを引起こす。制御ロジック２９１は信号を線Ｌに置いて、マルチプレクサ２９４が、復号されたデータの３２ビットワードをバスＤへ送ることを引起こす。制御ロジック２９１はさらにバスＧ上にアドレス信号を置いてフレームデータ入力レジスタＦＤＲＩをアドレス指定する。次のクロック信号は復号されたデータをバスＥに移し、ここでこれはフレームレジスタにロードされ、フレームレジスタが一杯であれば最終的に構成メモリ１２内へ、ＦＡＲレジスタで示されたアドレスにシフトされる。

【００５７】

ｍｏｄＣＢＣ値は復号動作中ただ１回だけ用いられる。暗号化されたデータの後続の６４ビットワードは、復号されてから、ＸＯＲ演算のため先に復号されたデータを用いて連鎖される。（ＦＡＲレジスタに記憶された値もまた、一度だけフレームアドレスの選択に用いられる。この後、フレームアドレスはフレームが一杯になるごとに単に増分される。）

【００５８】

動作の流れ

図８は、構成ロジック２９および復号器２４により行なわれる動作の流れを示す。最初にステップ７０において、構成ロジック２９はビットストリームヘッダをロードし、対応するデータを図４ｂに示す構成ロジックレジスタに置き、これにはビットストリーム長の判断も含まれる。ステップ７１で、準備シーケンスのさらなる部分として、構成ロジック２９は１番目の構成メモリアドレスを読出す。ビットストリームフォーマットは、暗号化が用いられているかどうかを示すオペコードを含んでいることを想起されたい。ステップ７２はオペコードの値による分岐である。暗号化が用いられていなければ、プロセスは図８の左部分に示されたものとなる。暗号化が用いられていれば、プロセスは図８の右に示されたものとなる。暗号化がない場合、ステップ７３で、構成ロジック２９はカウンタをビットストリームワードカウント（図４ｃを参照）に等しく設定する。ステップ７４で、構成データの３２ビット（１ワード）が構成メモリ１２のアドレス指定されたフレームへ送られる。ステップ７５で、カウンタが終了していないことが示されれば、ステップ７６でカウンタは減分され、構成データにおける次の１ワードが構成メモリ１２に送られる。カウンタが終了していれば、構成ロジック２９はクリーンアップ作業を行なうが、これには、最終的な循環冗長値を読出してビットストリームの終わりにある値と比較することによって、ビットストリームのロードにエラーがあったかどうかを判断することが含まれる。

【００５９】

ステップ７２で、ビットストリームが暗号化されていることが示されれば、カウンタにはワードカウントがロードされ、プロセスのステップ８１で、鍵アドレスレジスタ２９３（図６）からの初期鍵アドレスが復号器２４にロードされる。

【００６０】

ステップ８２で、暗号化された構成データの２つのワード（６４ビット）が復号器２４にロードされる。ステップ８３で、アドレス指定された鍵が復号器２４にロードされる。一実施例では、６４ビットの数が復号器２４にロードされる。この６４ビットの数は、５６ビットの鍵と、これが最初、中間、最後、または唯一の鍵であるかを示す２つのビットと、その他のビットとを含み、その他のビットは使用されないことも、パリティに使用されることも、または別の目的に使用されることもある。別の実施例では、６４ビット鍵データは、これが最後の鍵であるかどうかを示す単一のビットを含む。さらに別の実施例では、６４ビット鍵データは次の鍵についてのアドレスを含み、このため鍵を逐次的な順序で用いる必要はない。別の実施例では、付加的なビットがなく、鍵データが使用するビットは６４個を下回る。さらに別の実施例では、鍵でなくビットストリームがいくつの鍵を用いるべきかを示すが、これは安全性がより低いと考えられており、なぜなら攻撃者はいくつの鍵が用いられているかを知って、一回につき１つの鍵を破る単一の鍵攻撃を行なうことができるからであり、これに対し、いくつの鍵を用いるべきかを示すために鍵を用いれば、この情報が攻撃者に漏れることはない。

【００６１】

ステップ８４で、復号器２４はたとえばＤＥＳアルゴリズムを用いて６４ビットデータを５６ビット鍵で復号する。ＤＥＳアルゴリズムはブルース・シュナイヤーによる上述の書の２６５頁から２７８頁に記載されている。他の暗号化アルゴリズム、たとえば高度暗号規格ＡＥＳを用いることもできる。他のアルゴリズムはより多くの鍵ビットを必要とすることがあり得る。たとえばＡＥＳは１２８ビットから２５６ビットの鍵を必要とする。

【００６２】

ステップ８５で、より多くの鍵を用いるべきかどうかが判断される。鍵が最初、中間、最後、または唯一の鍵であるかを示す２つのビットを調べて、これが最後の鍵であるかどうかを判断し、もしそうでなければ鍵アドレスは増分され、復号器２４はメモリ２３内で次の鍵をアドレス指定する。

【００６３】

最後の鍵が用いられた後、ステップ８７で、レジスタＦＡＲおよびｉｎｉｔＣＢＣを組合わせて得られた６４ビット値として図６に示した、変更を加えられたＣＢＣ値と、ステップ８７で入手された復号された値とのＸＯＲが取られる。一実施例では、ＣＢＣレジスタ内にロードされた６４ビットランダム数のうち２２ビットは、ビットストリームの始めのフレームアドレスと交換される。暗号化プロセスの目的は、６４ビットの暗号化された値におけるあらゆる２進数を、すべての先行のビットおよび鍵の関数にすることである。ＣＢＣ値を１番目のアドレスと組合わせる目的は、意図された開始アドレスとは異なるアドレスにビットストリームがロードされると、暗号化された値が変わるようにすることである。ステップ８７はこれら両方の目的を達成する。次に新たなＣＢＣが記憶される。記憶は図６に示すＦＡＲおよびｉｎｉｔＣＢＣレジスタであっても、または復号器２４内に位置する別のレジスタであってもよい。

【００６４】

ステップ８８で、この復号された構成データはバス２７（図３）で構成ロジック２９へ送られる。構成ロジック２９は更新された巡回冗長検査値を算出し、これはロードのプロセスの終わりにＣＲＣレジスタに記憶された巡回冗長値と比較される。構成ロジック２９が暗号化を使用するように設定されていれば、構成ロジック２９内のマルチプレクサは、この復号された構成データを構成メモリ１２のアドレス指定された列へ送る。

【００６５】

ステップ８９でカウンタが調べられ、もし終了していなければステップ９６でカウンタは減分され、プロセスはステップ８２へ戻り、ここで次の６４ビット（２ワード）がビットストリームからロードされる。

【００６６】

最後に、ステップ８９でカウンタが終了したことが示されれば、ステップ９０で、ビットストリーム内のＣＲＣ（巡回冗長検査）値が、ビットストリームのロードの際に算出されるＣＲＣ値と比較される。値が一致すれば、構成は完了しており、ＦＰＧＡは動作へ移る。値が一致しなければ、ロードにエラーが発生しており、構成プロセス全体が中断される。

【００６７】

鍵順序の評価－単一の鍵攻撃の防止

図９は、鍵順序を評価するために復号器２４により実現される状態機械を示す。Ｅｎｃ＿ｄａｔａ＿ｒｅａｄｙ信号が活性化されるまで状態機械は状態Ｓ１に留まる。この信号は復号を始めてもよいことを示し、判断状態Ｑ１へ移らせ、ここで復号器２４は、バス２７上のアドレスＩｎｉｔ＿ｋｅｙ＿ａｄｄｒにより特定されるアドレスをバス２８に与え、鍵および鍵順序をリードバックし、さらに鍵順序データの２ビットから、鍵が１番目の鍵または唯一の鍵であるかどうかを判断する。もしそうでなければ、復号器２４はＢａｄ＿ｋｅｙ＿ｓｅｔ信号を制御ロジック２９１に送り、構成ロジック２９に構成を中断させる。アドレスが１番目または唯一であれば、復号器２４は状態Ｓ３に移り、ここでデータが復号される。次に状態機械は判断状態Ｑ２に移り、ここで鍵が最後または唯一であるかどうかが判断される。もしそうであれば、復号は完了しており、状態Ｓ４で復号器２４は復

号されたデータを構成ロジック２９に返す。もしそうでなければ、状態Ｓ５で、復号器２４は鍵アドレスを増分して新たな鍵を得る。状態機械は質問Ｑ３を問い、次の鍵が中間の鍵または最後の鍵であるかどうかを判断する。もしそうでなければ、Ｓ２は構成を中断させる。鍵が中間または最後であれば、状態機械は状態Ｓ３に戻り、データを再び復号する。別の実施例では、状態Ｓ４で復号器２４はさらに、復号されたデータとＣＢＣ値とのＸＯＲをとるステップを行なう。

【００６８】

鍵順序を鍵内で記憶する利点は、攻撃者が単一の鍵攻撃を実行できないことであり、それは、復号器２４が復号を行なう際に（設計者が意図したように）鍵メモリ２３で特定された鍵すべてを使用することを、攻撃者は妨げることができないからである。単一の鍵攻撃を用いている攻撃者から保護するためには、２番目および３番目の質問Ｑ２およびＱ３を問う必要はないが、それは鍵順序がＰＬＤの内部にある鍵データ内に記憶されるからである。しかし、各々の鍵がロードの際に正しくラベル付けされたことを確かめるために、設計者、または鍵をロードするボードの検査員がこれら３つの質問を問うことは有益である。

【００６９】

一実施例で復号器２４は、復号－暗号化－復号の順番をとる三重ＤＥＳ規格を用い、別の鍵が用いられるたびにアルゴリズムを（僅かだけ）交替させる。このような組合せは、ＡＮＳＩ　Ｘ９．５２　１９９８三重ＤＥＳ規格に準拠する。別の実施例ではその都度復号が用いられる。

【００７０】

鍵メモリ２３

図１０ａに示す回路は３つの構成要素、すなわち電池電源スイッチ２２、制御ロジック２３ａおよび鍵レジスタ２３ｂを含む。制御ロジック回路２３ａおよび鍵レジスタ２３ｂは図３の鍵メモリ２３を成す。図１０ａの実施例では、鍵レジスタ２３ｂは６つの６４ビットワードを含む。当然のことながら、これに代えて他の鍵メモリサイズを用いてもよい。実施例によっては、鍵メモリ２３には６個をはるかに上回る鍵が記憶され、使用すべき鍵のアドレスを与えるためには４ビット以上が必要である。鍵レジスタ２３ｂ用の電源は電池電源スイッチ２２から線ＶＳＷＩＴＣＨを通って来る。鍵メモリ電源電圧ＶＣＣＩが不十分、またはない場合、電池電源スイッチ２２は電池バックアップ電圧ＶＢＡＴＴをＶＳＷＩＴＣＨ線に印加し、こうしてＶＳＷＩＴＣＨは正の電圧を伝える。

【００７１】

この実施例では、各々の鍵レジスタは６４個のメモリセルを有する。各セルは書込イネーブル信号ＷＥを受取り、この信号はハイであればセルへのデータの書込を引起こし、ローであればセル内のデータの保持を引起こす。書込イネーブル信号ＷＥは１レジスタ内のセルに共通である。ＰＬＤ電源電圧（ＶＣＣＩとは異なる）がなくＷＥ信号が活性状態に駆動されない場合、Ｔ１などの弱いプルダウントランジスタがＷＥ信号をプルダウンし、こうして鍵メモリレジスタをいずれもアドレス指定不可能にし、メモリセルのいずれも乱されないようにする。

【００７２】

一実施例では、復号鍵をＰＬＤ内にロードするためにＰＬＤのＪＴＡＧポートを用いる。メモリセル電源電圧は通常の動作時にはＶＣＣＩのデバイス電圧レベルにあり、一実施例でこのレベルは３．０ボルトから３．６ボルトの間である。ＪＴＡＧポートに与えられる信号はいくつかの異なった電圧であり得る。また、いくつかの異なった内部電圧があり得る。したがって電圧変位が必要とされる。この電圧変位はメモリセルで行なわれる。メモリセルの詳細を図１０ｂで示す。インバータＩ１およびＩ２を含むラッチには、ＶＳＷＩＴＣＨで電力が与えられるため、デバイス電源電圧ＶＣＣＩがあるかないかにかかわらず電力が与えられる。ＷＥ信号および反転されたデータ信号ｄａｔａ＿ｂの両方は１．５ボルトのレベルで動作する。これら信号はＮＭＯＳトランジスタＴ４、Ｔ５およびＴ６を駆動し、さらにインバータＩ３を通じて（やはり１．５ボルトの電源電圧を用いて）トラン

ジスタＴ７を駆動する。図１０ｂは、ＷＥがローであれば、トランジスタＴ４およびＴ５がオフであり、インバータＩ１およびＩ２を含むラッチの内容が維持されることを示している。ＷＥがハイであれば、インバータＩ１およびＩ２のいずれかがローにされ、こうして新たなデータをラッチ内にロードする。

【００７３】

制御ロジック回路２３ａはＪＴＡＧバス２５（図３にも示す）から信号を受取る。ＪＴＡＧバス２５は、書込、読出、安全モードの設定のための制御信号、ならびにデータおよびアドレスバスを含む。このインターフェイスはＩＥＥＥ１５３２ＪＴＡＧ規格に準拠する。鍵メモリ２３がＪＴＡＧバス２５によってアクセスされ得る前に、安全性状態（バス２６）が非安全モードに置かれるが、これはＩＳＣ＿ＰＲＯＧＲＡＭ＿ＳＥＣＵＲＩＴＹ命令（図１０ａを参照）を用い、鍵データバスのビット０に論理１を与えることによって行なわれ得る。鍵メモリ２３は、ＩＥＥＥ１５３２規格のＩＳＣ＿ＰＲＯＧＲＡＭおよびＩＳＣ＿ＲＥＡＤ命令を用いて、ＪＴＡＧバス２５に書込まれ、およびここから（検証のために）読出される。制御ロジック２３ａは、ＪＴＡＧバス２５からの３ビットアドレス信号ＡＤＤＲをデコードするためのデコーダを含み、ＪＴＡＧバス２５にＩＳＣ＿ＰＲＯＧＲＡＭ命令が現われれば書込ストローブ線ｗｓ＿ｂ［５：０］のうちアドレス指定された線上にローへのパルスを生成、またはＩＳＣ＿ＲＥＡＤ命令がＪＴＡＧバス２５上に現われれば読出選択線ｒｓｅｌ［５：０］のうちアドレス指定された線上にハイの信号を生成する。６つの６４ビットワードのうち１つの読出は、６本の読出選択線ｒｓｅｌ［５：０］のうち１本にハイの信号を与えることで可能であり、これは読出マルチプレクサ２３ｄが、６４本の出力線ｑ［６３：０］上に選択されたワードを置くことを引起こす。書込選択線または読出選択線のうち１本のみが一度に選択される。読出選択線がアサートされなければ、ハイのｐａｒｋ＿ｌｏｗ信号は、６４個のトランジスタ２３ｅが６４線ｑ［６３：０］をプルダウンし、これら線のフローティングを防ぐことを引起こす。

【００７４】

鍵メモリ２３が非安全モードで動作していれば、６４ビットワードは鍵レジスタ２３ｂからＪＴＡＧバス２５へ読出され、ここで値はＦＰＧＡの外部で調べられ得る。ＦＰＧＡはこの非安全モードにおいて、ＤＥＳ復号のための５６ビット鍵としてレジスタ２３ｂ内の選択された６４ビットワードのうち５６ビットを用いることによって試験され得る。一実施例で、鍵メモリ２３が非安全モードにあれば、設計がロード前に暗号化されていてもユーザの設計のリードバックは可能である。これによって設計者は、暗号化された設計も試験およびデバッグできる。鍵の安全性状態の通信はバス２６（図３も参照）を通じて行なわれる。

【００７５】

値が鍵レジスタ２３ｂ内に書込まれ、バス２５からの読出動作で検証された後、ＩＳＣ＿ＰＲＯＧＲＡＭ＿ＳＥＣＵＲＩＴＹ命令を用いて、ＩＥＥＥ１５３２規格の一部である６４ビット鍵データバスのビット０に論理０を与えることによって、制御ロジック２３を安全モードに置く。この安全モードでは、鍵に対するアクセスは認められない。

【００７６】

図１１に示すように、攻撃者がＩＳＣ＿ＰＲＯＧＲＡＭ＿ＳＥＣＵＲＩＴＹ命令を用いてから鍵を読出すことによって非安全モードに戻ることを確実に防ぐために、安全性がなくなると（すなわちＩＳＣ＿ＰＲＯＧＲＡＭ＿ＳＥＣＵＲＩＴＹ信号が非安全論理レベルに移ると）制御ロジック２３ａ内の状態機械は０を６つのワードすべてに、一回につき１ワードずつ書込むことで、すべての鍵を消去する。これは以下のステップ１１０からステップ１１９によって行なわれる。ステップ１１０で、０をｗｄａｔａ［６３：０］バスに置き、ステップ１１１でｗｓ＿ｂ［０］信号を（論理０値で）アサートし、次にステップ１１２～１１７で、ｗｓ＿ｂ［０：０］からｗｓ＿ｂ［５：０］までの信号を一回につき１つずつ逐次的にストローブし、この後ステップ１１８で安全性状態を変えて非安全モードに入り、最後にステップ１１９でｗｄａｔａ［６３：０］論理０値を解放する。このように、電池でバックアップされるメモリ２３を非安全モードに置くあらゆる試みは、鍵レジ

スタ２３ｂ内の値すべての消去を引起こす。

【００７７】

鍵メモリ２３が安全モードにあるかどうかを通信するため、制御ロジック２３ａはバス２６（これは単一の線であり得る）上で安全モード信号を構成ロジック２９に送り、鍵メモリ２３が安全モードで動作していることを示す。この信号が非安全モードに切換わると、構成ロジック２９は設計を構成メモリ１２からクリアする。鍵が鍵レジスタ２３ｂに記憶され、かつ鍵メモリ２３が安全モードにあっても、暗号化されていないビットストリームが構成ロジック２９により構成メモリ１２内にロードされ得ることに注目されたい。

【００７８】

鍵のロード、多数の暗号鍵

ユーザが設計の詳細を知ることができない安全モードにＰＬＤが置かれる前に、復号鍵をＰＬＤ内にロードする必要がある。図３に示す実施例では、鍵はＪＴＡＧポート２０を通じてロードされる。

【００７９】

この発明の特徴として、暗号鍵はこのＪＴＡＧポート２０を通じてロードされる。ＪＴＡＧプログラマがボード試験中に暗号鍵をロードすることが予測される。鍵を記憶するためのＲＡＭが非安全モードにあれば、ユーザはこれに対する完全なアクセスを有し、設計が暗号化されていたとしても鍵および設計の両方を読出すことができる。このことは鍵と鍵の使用とを試験している設計者にとって有用である。次に設計者は動作に一旦満足すると、ＪＴＡＧポートを通じ別の命令を送って鍵メモリを安全モードに置くことができる。鍵メモリが一旦安全モードに置かれると、鍵を読出すことはできない。さらに、鍵メモリを安全モードから非安全モードに移すと、メモリ初期化プロセスを開始する回路が活性化され鍵は消去される。（後に論じる図１５で、この機能を行なうための状態機械を示す。）

【００８０】

この発明の一局面に従うと、設計を暗号化するために２つ以上の鍵が用いられ得る。たとえば３つの鍵を用いる場合、ビットストリームはまず１番目の鍵を用いて暗号化され、次に結果として得られる暗号化されたビットストリームが２番目の鍵を用いて再び暗号化され、最後に、結果として得られる二重に暗号化されたビットストリームが３番目の鍵を用いてさらに暗号化される。この三重に暗号化されたビットストリームは、たとえばＰＬＤを保持するプリント回路基板上のＰＲＯＭまたはフラッシュメモリで記憶される。

【００８１】

復号では、これら鍵を（逆の順序で）連続的に用いて、暗号化されたビットストリームを反復的に復号する。これに加え、特定の設計を復号するのに用いられるよりも多くの鍵がＰＬＤに記憶されていれば、暗号化されたビットストリームは暗号化されていない部分の中に、いくつの鍵を用いるべきかの指示と１番目の鍵のアドレスとを含み得る。このような実施例では攻撃者によるビットストリームの解読が容易となり得るが、それは攻撃者が一度に１つの鍵に対処するだけでよいからである。これに代えて、鍵自体が、最初、中間、最後、または唯一の鍵であるかを示すこともある。こうして、同じＰＬＤが異なったときにプログラムされて、（異なった設計で構成される）異なった機能を行なうことが可能となり、異なった鍵の値についての情報を、設計者のうちただ一人または数人にとってのみ利用可能とできる。こうして第１の設計者は第２の設計について、両方の設計が同じＰＬＤで（異なったときに）実現されたとしても知らないことがあり得る。

【００８２】

図３を参照して、構成ロジック２９は図１の構成ロジック１４の他に追加のロジックを含む。図１の構造では、構成アクセスポート２１上のビットストリームはワード、一実施例では３２ビットワードとして扱われる。これらワードのうち、通常はビットストリームの初めまたはその近くにあるいくつかのワードは、たとえばビットストリームの長さ、構成データについての開始アドレスなどのヘッダ情報を含む。この発明のビットストリームで新規な点は、ビットストリームが暗号化されているかどうかについての指示、およびビットストリーム内の構成データを復号するための鍵のアドレスである。

【００８３】
電池でバックアップされるメモリ
鍵メモリ２３に記憶される値は、ＦＰＧＡへの電力が外されたときには電池によって維持されることが好ましい。
【００８４】
さらに、スイッチ２２などの電池電源スイッチを用いて、暗号鍵以外のメモリもまたバックアップできる。具体的に、ＰＬＤの電源切断時にＰＬＤの生成したデータを保存することが目的であれば、ＰＬＤ内のフリップフロップすべてにＶＳＷＩＴＣＨ電圧供給が経路付けられるＰＬＤを製造することができる。さらに、ＰＬＤの電源切断時にＰＬＤの構成を保存することもまた目的であれば、構成メモリ１２（図３）は代替的にＶＳＷＩＴＣＨから電力を与えられ得るが、このような実施例では、単にＰＬＤ内のフリップフロップに電力を与えるよりもかなり多くの電池電力が必要となり、さらにフリップフロップに電力を与えるには、数個の暗号鍵を記憶するための極めて小型のメモリに電力を与えるよりも多くの電池電力が必要となる。
【００８５】
図１２は電池電源スイッチ２２の構造を示す。この実施例では、ＶＢＡＴＴレベルシフト回路３１によって、ＰＬＤが電池および主電源につき異なった電圧を用いることができる。さらに、当然のことであるが、この回路の目的はさまざまな電圧レベルに対処することである。一実施例で電池電源スイッチ２２は最大３．６ボルトのＶＣＣＩ電圧を扱うことができ、ＶＣＣＩが約１ボルト未満に落ちると電池電力へ切換わる。電池電圧は１．０ボルトから３．６ボルトの間であり得る。
【００８６】
電池電源スイッチ２２は４個の出力駆動Ｐチャネルトランジスタ Ｐ０～Ｐ３を含む。トランジスタＰ０およびＰ１は一緒にオンおよびオフになり、トランジスタＰ２およびＰ３も同様である。この回路は、ＶＣＣＩおよびＶＢＡＴＴが相互に接続されてしまうあらゆる可能性を回避するために、各々の脚につき１個でなく２個のトランジスタを含む。トランジスタＰ０は寄生ダイオード（ドレインと基板とのｐ－ｎ接合）を含み、これはトランジスタがオフである際にも図で上方向へ電流を伝導するおそれがある。このような電流の流れを防ぐためにトランジスタＰ１が追加され、その基板はそのドレインに接続されるため、寄生ダイオードの伝導は下方向でのみ可能である。同様の仕組みをトランジスタＰ２およびＰ３にも設ける。したがって、電流がＶＢＡＴＴからＶＣＣＩへ、またはＶＣＣＩからＶＢＡＴＴへ伝導する可能性はない。インバータ３３および３４はＶＳＷＩＴＣＨ電圧から電力を与えられるため、ＶＣＣＩがオフであっても常に動作している。トランジスタＰ４は常にオンの抵抗器であり、静電放電に対する保護を提供する。ほとんどの時間、トランジスタＰ４により制御されるこれら構造は電流を引込まず、通常トランジスタＰ４にわたって電圧の降下はない。
【００８７】
図１３はＶＢＡＴＴレベルシフト回路３１の一実施例を示す。端子ＯＵＴにおける出力電圧は信号ＩＮおよびＩＮＢにより制御される。これら信号はインバータ３３および３４により生成され、これらインバータはＶＳＷＩＴＣＨノードから自分の電源電圧を得る。したがって、ＶＳＷＩＴＣＨがＶＢＡＴＴにより供給される場合、信号ＩＮおよびＩＮＢのうち一方は電圧ＶＢＡＴＴにあり、他方は接地される。しかし、もしＶＳＷＩＴＣＨがＶＣＣＩにより供給される場合、ＩＮおよびＩＮＢのうち一方はＶＣＣＩの電圧レベルにあることになる。ＩＮがＶＣＣＩにあり、かつＩＮＢが接地されている場合、トランジスタ４５はオンであり、トランジスタ４６はオフである。Ｐチャネルトランジスタ４３のゲートはローであり、トランジスタ４３はオンであり、こうしてインバータ４７の入力をＶＢＡＴＴにする。トランジスタ４８の出力もまたＶＢＡＴＴにある。再び図１２を参照して、トランジスタＰ０のゲートでの電圧レベルＶＢＡＴＴはトランジスタＰ０を正にオフにする。
【００８８】

図１４はＶＣＣＩ検出回路３２を示す。ＶＣＣＩ検出回路３２は、線ＶＳＷＩＴＣＨでの電圧がいつ電池に切換えられ、いつ再びＶＣＣＩに切換えられるのかを判断する。回路３２のこの実施例は本質的に一続きになった５個のインバータ段Ｉ１からＩ５である。スイッチング電圧の制御は主にインバータ段Ｉ１で行なわれる。トランジスタ５２および５３はＣＭＯＳインバータを形成する。このＣＭＯＳインバータへの電力はＰチャネルトランジスタ５１を通じて流れることになり、トランジスタ５１は、ＶＣＣＩがトランジスタ５１のしきい値電圧、典型的に０．７～０．８ボルトに達するまでオンにならない。ＶＣＣＩのスイッチングが遅く、一杯の電圧に達するまで数ミリ秒かかる場合、トランジスタ５１は回路Ｉ１の活性化を遅らせる。トランジスタ５１がオンになると、トランジスタ５２のソース（上の端子）はＶＣＣＩになる。Ｎチャネルトランジスタ５３は典型的に、やはり約０．７～０．８ボルトのしきい値電圧を有するが、トランジスタ５２に対して弱いトランジスタとしてのサイズにされる。一実施例でトランジスタ５３の幅／長さ比は１／１８であるのに対し、トランジスタ５２の幅／長さ比は３／２である。したがってトランジスタ５３は、トランジスタ５２がオンになるまでインバータＩ２の入力をローにするにすぎない。一実施例で回路Ｉ１は、ＶＣＣＩが約１．０ボルトにあるときにインバータ段Ｉ２の入力をハイにする。したがってインバータ５４の出力はローになる。インバータ段Ｉ３はシュミットトリガである。インバータ段Ｉ３への０ボルトの入力はトランジスタ５６および５７をオフにし、トランジスタ５５をオンにし、ノードＮ３をＶＣＣＩにし、トランジスタ５８をオンにし、これはノードＮ４を引上げ、こうして、トランジスタ５６がオンになる電圧を上昇させ、ＶＣＣＩの小さな変動がノードＮ３での電圧を切換えることを防ぐ。インバータ５９および６０は任意であり、出力信号ｕｓｅｂａｔｔおよびｕｓｅｂａｔｔｂのより鋭いエッジをもたらし、これにより図１２の電池電源スイッチ２２がＶＢＡＴＴからＶＣＣＩに切換わることを引起こす。ＶＢＡＴＴ′信号が制御するトランジスタ６１は弱いプルダウンのトランジスタであり、ＶＣＣＩがなくインバータ６０から出力信号をもたらさないときにｕｓｅｂａｔｔｂ線を確実にローにする。

【００８９】

構成されたＰＬＤを含む製品の購入者に利用可能でない鍵

ＰＬＤを構成するのに用いられた設計を攻撃者が知ることを防ぐため、いくつかの追加のステップをとることがある。

【００９０】

別の局面に従うと、ＰＬＤを組込んだシステムの販売前に鍵がＰＬＤ内にロードされ、これにより、ＰＬＤを含むシステムの販売後、設計をＰＬＤ内にロードして使用できるが、攻撃者は鍵に記憶された値を知ることはできない。したがって、暗号化されていない設計は読出またはコピーされ得ない。この安全性を達成するためにいくつかのステップがとられる。

【００９１】

安全モード保存（改ざん防止）

一実施例で、ＰＬＤの構成ロジック２９には安全性に関する２つのフラグがある。一方は、復号鍵が安全にされているかどうかを示し、他方は、設計が復号された設計であり、したがって保護されなければならないことを示す。ＪＴＡＧロジック１３（図３）がＩＳＣ＿ＰＲＯＧＲＡＭ＿ＳＥＣＵＲＩＴＹ命令で安全モードを選択すると、制御ロジック２３ａ（図１０ａ）にあるｓｅｃｕｒｅ＿ｋｅｙフラグがセットされる。ビットストリーム内の設計データが暗号化されているということを、ＰＬＤ内にロードされたビットストリームが示している場合、構成ロジック２９（図示せず）にあるｓｅｃｕｒｅ＿ｄｅｓｉｇｎフラグがセットされる。いずれかのフラグが後に非セットにされると構成メモリ全体がクリアされ、復号された設計がなくされる。ｓｅｃｕｒｅ＿ｋｅｙフラグが（ＩＳＣ＿ＰＲＯＧＲＡＭ＿ＳＥＣＵＲＩＴＹ命令により）リセットされると、鍵もまた消去される。

【００９２】

図１５は、設計クリア機能を行なう状態機械を示す。ｓｅｃｕｒｅ＿ｄｅｓｉｇｎフラグがセットされると、状態機械は状態Ｓ１に入る。この状態は、ｓｅｃｕｒｅ＿ｄｅｓｉｇ

ｎフラグの安全モードから非安全モードへの変化を監視する。設計安全（ｓｅｃｕｒｅ－ｄｅｓｉｇｎ）モードが継続する限り状態機械は状態Ｓ１に留まる。変化が一旦生じると、状態機械は状態Ｓ２に入り、データを構成メモリ１２内にシフトするためのデータシフトレジスタはリセットされ、こうして構成メモリビットについてのデータ線すべてに０を置く。次に状態機械は状態Ｓ３に移り、アドレス指定されたフレームのワード線がアサートされる。この結果、データシフトレジスタ線上の０は、アドレス指定されたフレームにあるメモリビット内に書込まれる。質問Ｑ１で、アドレス指定すべきフレームがまだあることが示されると、状態機械は状態Ｓ４に移り、ここでフレームアドレスが進められ、状態機械は状態Ｓ３に戻る。質問Ｑ１で、アドレス指定すべきフレームがもうないことが示されると、プロセスは終了し構成メモリはクリアされる。

【００９３】

鍵を攻撃者によるアクセスから保護することもまた必要である。鍵のロードは、設計を含むシステムが最終顧客にとって利用可能となる前に行なわれる。設計者は設計の開発中、ＰＬＤをデバッグのために非安全モードで動作させたいと望むことがある。このデバッグ動作を可能にし、かつ鍵の安全性を保存するため、鍵レジスタすべてのクリアで、鍵ロードのプロセスが非安全モードで始まる。鍵のロード中、および鍵が検証のためリードバックされる間、安全な鍵のフラグが非安全モードに保たれることになる。この安全鍵フラグはまた、構成ビットストリームがロードおよび復号される間も非安全モードに保たれる得る。しかし安全鍵フラグが一旦セットされると、安全鍵フラグを非安全モードに戻せばすべての鍵がクリアされ、さらに図１５の状態機械の動作が開始される。したがって鍵がクリアされるばかりでなく、構成もまたクリアされる。

【００９４】

リードバック攻撃、およびリードバックの禁止

いくつかのＦＰＧＡはＦＰＧＡからのビットストリームのリードバックを許しており、このためユーザは設計をデバッグしたり、またはＦＰＧＡ内のフリップフロップから状態機械情報を入手できる。設計がリードバック動作用に再暗号化されない限り、ビットストリームをリードバックする行為によって、暗号化されていないビットストリームが可視となるよう暴露されてしまう恐れがある。

【００９５】

設計のさらなる安全性は、暗号化された設計がＦＰＧＡにロードされる際にリードバックを禁止することで提供される。一実施例でリードバックは、復号鍵もまた安全にされる場合にのみ禁止される。

【００９６】

図１６は、構成メモリをロードおよびリードバックするための構造のブロック図を示す。一実施例で構成ロジック２９は以下の２つの条件が揃っている場合にリードバックを防止する。すなわち、（１）データバス２６（図３および図１０を参照）上の安全性状態線によって、鍵が安全モードにあることが示されている、および（２）ビットストリームが暗号化されていることを示す構成ビットストリーム内のオペコードに構成ロジック２９が応答したことがある、の２つの条件である。したがって、鍵が安全にされていない、またはビットストリームが暗号化されていない場合、リードバックは許可され得る。別の実施例では、異なる条件によって、リードバックが許可され得るかどうかが制御される。

【００９７】

構成ロジック２９は、リードバックが行なわれるであろうことを示すヘッダをビットストリーム内で受取ると、そのフレームアドレスレジスタに記憶されたフレームアドレスを線１０７上に送り、このフレームアドレスはアドレスデコーダ１１０によってデコードされてバス１０９のアドレス指定された線を選択する。次に、線１０８上のワード線イネーブル信号がアサートされ、この信号はバス１０９の選択されたワード線をアサートし、こうして、選択されたワード線によりアドレス指定されたメモリセルは、その値をｎ本のデータ線１０２上に置く（ｎはフレーム長であり、構成ロジック２９内に記憶される）。次に構成ロジック２９は線１０４上でロード信号をアサートし、データのフレームを（並列に

）データシフトレジスタ１０１内にロードする。次に構成ロジック２９は線１０５上でシフト信号をアサートし、データシフトレジスタ１０１が３２ビットワードのデータのフレームをバス１０３上でフレームデータ出力レジスタ（図４ｄを参照）へ、さらにそこから構成アクセスポート２１（図３）上で出ていくビットストリームへシフトすることを引起こす。

【００９８】

ビットストリームに復号が示されている場合、構成ロジック２９は内部フラグをセットしてこれを示す。これらフラグがセットされ、かつバス２６上の安全性状態信号により示されるように鍵メモリ２３が安全モードにあれば、構成ロジック２９は、線１０８上のワード線イネーブル信号を非活性に保ち、かつ線１０４および１０５上のロードおよびシフト信号を非活性に保つことで、ビットストリーム内のリードバックコマンドに応答してリードバックを防ぐ。しかし鍵メモリ２３が安全モードになければ、設計が暗号化されているかもしれないにもかかわらず、リードバックが許され、試験およびデバッグが可能となる。

【００９９】

部分的再構成攻撃およびその防止

いくつかのＦＰＧＡはＦＰＧＡの部分的再構成を許すか、または、別個の開始アドレスおよび別個の書込命令を用いてＦＰＧＡの異なった部分に設計の異なった部分をロードすることを許している。攻撃者は設計を知る試みとして設計を部分的に再構成し、ブロックＲＡＭまたはフリップフロップの内容を出力ポートに直接読出したり、または既存の設計に或るセクションを追加して、設計を知るのに用いられ得る情報を読出すことがあり得る。たとえば攻撃者は、暗号化されていない設計であって、その唯一の目的は暗号化された設計についての情報を抜き出すことである、暗号化されていない設計によって部分的にＰＬＤを再構成することがあり得る。このようなトロイの木馬設計は別のビットストリームでＰＬＤ内にロードされたり、または既存の暗号化されたビットストリームに加えられたりする。たとえばＦＰＧＡのブロックＲＡＭ内にロードされた状態機械設計を知ることが攻撃者の関心であれば、トロイの木馬設計はブロックＲＡＭのアドレスを通じて巡回し、ブロックＲＡＭデータ内容をパッケージのピンに送るためのロジックを含み得る。

【０１００】

攻撃者がこのような変化を加えることを防ぐため、元の設計が暗号化される場合、復号を伴う構成が一旦始まると構成ロジック２９は部分的な再構成を却下する。構成ロジック２９は、復号オペコードを有するヘッダが一旦処理されるとさらなる書込命令を却下する。さらに構成ロジック２９は、暗号化を伴わない構成が一旦行なわれると、復号を伴う構成を却下する。構成ロジック２９はこれらの制限を達成するために、復号命令が受取られた後に構成メモリに書込を行なうヘッダを無視し、さらに設計の暗号化されていない部分がロードされた場合に復号コマンドを有するヘッダを無視する。こうして、復号を伴う書込が用いられていることをオペコードのうちいずれかが示すと、ＰＬＤは単一の書込命令のみを受入れることになる。

【０１０１】

さらなる実施例

これら図面に関する以上の記載はいくつかの実施例についての詳細を与えている。しかし多くの追加的な実施例もまた可能である。たとえば、上述の暗号ブロック連鎖アルゴリズムの代わりに、たとえば一回につき１つの８ビットバイトなどの、ブロックサイズよりも小さなユニットでデータが暗号化され得る、暗号フィードバックモードと呼ばれる暗号化方法を用いることが可能である。この暗号フィードバックモードはシュナイダーの同書２００頁から２０３頁に記載されている。

【０１０２】

さらに別の実施例では、暗号化が用いられると、あらゆるビットストリームはアドレス０で始まってロードされなければならない。この実施例の一実現例では、暗号化を特定するオペコードを受取ると、開始フレームアドレスレジスタＦＡＲ（図６）にロードされるア

ドレスをいずれもアドレス０と取替える。

【０１０３】

さらに別の実施例では、開始アドレスおよび設計データは両方とも暗号化される。この実施例では、暗号化されていない設計データで可能であるのと同様、異なったフレームアドレスで始まる暗号化された設計データのうちいくつかのセグメントをロードすることが可能である。

【０１０４】

別の実施例では、鍵メモリ２３などの鍵メモリに記憶される鍵データは、続く鍵の数を特定する。この実施例を変形したものでは、鍵データはさらにこの鍵に先行する鍵の数も特定する。設計者が意図した１番目の鍵アドレスとは違う鍵アドレスを攻撃者が与えると、構成は中断され得る。さらに、鍵内で特定された数の鍵が用いられるまで暗号化が進行する。

【０１０５】

別の実施例では、鍵メモリが非安全モードにある際に鍵をリードバック可能にする代わりに、鍵はパリティビットまたはＣＲＣチェックビットを含み、これらビットのみが、鍵が正しくロードされたことを検証するためにリードバックされ得る。この実施例では、１人の設計者が知っている鍵を別の設計者から秘密にしておくことができ、異なったときにＰＬＤを用いて異なった設計をロードする際にこの実施例は有用である。

【０１０６】

上述のＣＲＣチェックサム算出に関し、ＣＲＣチェックサムが設計の暗号化の前または後のいずれかに算出される実施例が提供され得る。当然のことながら、ビットストリームに追加されるチェックサムが設計データの暗号化前に算出される場合、その復号後に対応するチェックサムが設計データ上のＰＬＤ内で算出されることになる。同様に、ビットストリームに追加されるチェックサムが設計データの暗号化後に算出される場合、ＰＬＤは設計データの復号前に、受取ったビットストリーム上で対応するチェックサムを算出することになる。

【０１０７】

復号鍵をロードするプロセスに関するさらなる注としては、図８に例示したプロセスを用いる場合、復号鍵をロードするためにデバイスプログラマを用いる必要はない。鍵は単にボード試験手順の一部としてロードされ得る。

【０１０８】

さらに、上述の構造および方法を用いて２つ以上のＰＬＤをプログラムすることも可能である。いくつかのデバイスをデイジーチェーンに配置してこれら直列のデバイスを通じビットストリームを通過させるか、または直列のデバイスをアドレス指定することによって、単一のビットストリームを用いて２つ以上のＰＬＤまたはＦＰＧＡをプログラムすることが周知である。デバイスのうち１つまたはそれ以上が暗号化された設計データを受取る際、いくつかのＰＬＤをこのような配置に配置することが可能である。

【０１０９】

さらに別の実施例としては、暗号化された設計データを有するビットストリームにつき単一のアドレスのみが特定され得る一実施例を記載したが、別の実施例で、好ましくは暗号化されるいくつかのアドレスが、設計の別個の部分をロードするために特定され得る。さらにこれら別個の部分は同じ暗号鍵を用いても、異なった暗号鍵または異なった鍵の組を用いてもよい。

【０１１０】

以上の説明から明白となった変形例は、この発明の範囲内に含まれると意図されている。

【図面の簡単な説明】

【図１】先行技術のＦＰＧＡにおける機能関係を示す図である。

【図２】先行技術のビットストリームに含まれ得るビットストリームフォーマットおよびコマンドを示す図である。

【図３】この発明の一実施例に従うＦＰＧＡでの機能関係を示す図である。

【図４】この発明のビットストリームに含まれ得るビットストリームフォーマットおよびコマンドを示す図である。
【図５ａ】暗号化されていないビットストリームの例を示す図である。
【図５ｂ】暗号化されたビットストリームの例を示す図である。
【図６】構成ロジック２９と、復号器２４に通じるバス２７およびバス２８の線とを示す図である。
【図７ａ】この発明の一実施例で用いられる三重暗号化を伴う外部暗号ブロック連鎖における、変更を加えられた開始値を示す図である。
【図７ｂ】図７ａとともに用いられる、対応する開始値および復号プロセスを示す図である。
【図８】ビットストリームを処理するための動作を示すフローチャートである。
【図９】鍵順序を評価するために復号器２４により実現される状態機械を示す図である。
【図１０ａ】図３の鍵メモリ２３の構造を示す図である。
【図１０ｂ】図１０ａのメモリセルの構造を示す図である。
【図１１】非安全にされると鍵を消去するために図１０ａの制御ロジック２３ａが実行するステップを示す図である。
【図１２】図１０ａの電池電源スイッチをより詳細に示す図である。
【図１３】図１２の電池電源スイッチのレベルシフト回路を示す図である。
【図１４】図１２の電池電源スイッチの電圧検出回路を示す図である。
【図１５】安全モードから出ると設計を消去するための状態機械を示す図である。
【図１６】構成メモリをロードし構成をリードバックするための要素を示すブロック図であって、暗号化があるときにディセーブルされる線を含む図である。

【図１】

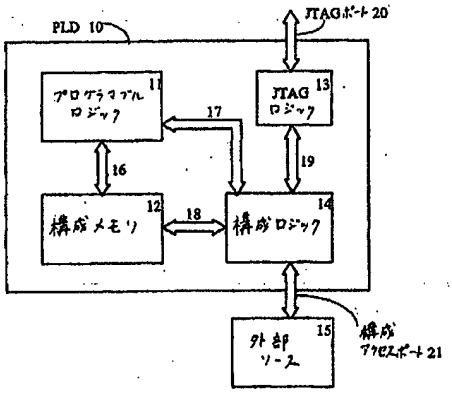


Fig. 1
PRIOR ART

【図２ａ】

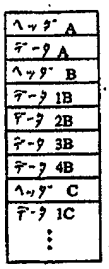


Fig. 2a
PRIOR ART

【図２ｂ】

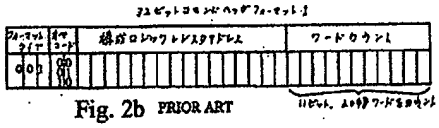


Fig. 2b PRIOR ART

【図２ｃ】

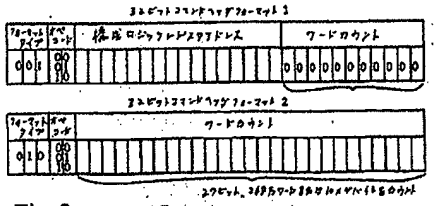


Fig. 2c PRIOR ART

【図２ｄ】

ビットストリームヘッダ構成ロジックレジスタアドレス	構成ロジックレジスタデータ内容
0000	巡回冗長検査（CRC）
0001	フレームアドレス
0010	フレームデータ入力
0011	フレームデータ出力
0100	コマンド
0110	制御
0111	状態
1000	デバイスID
1001	構成オプション
1010	予約
1011	フレーム長
1100	予約
1101	予約
1110	予約
1111	予約

Fig. 2d
PRIOR ART

【図３】

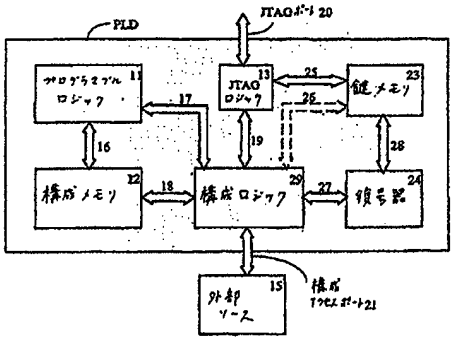


Fig. 3

【図４ａ】

Fig. 4a

【図４ｂ】

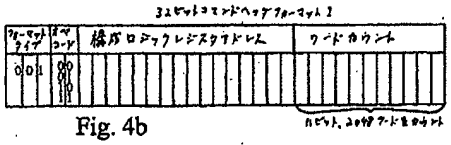


Fig. 4b

【図４ｃ】

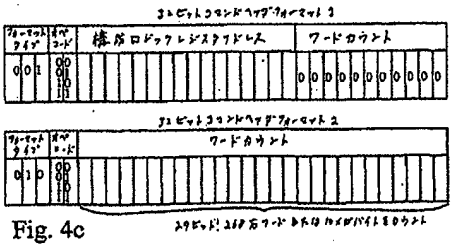


Fig. 4c

【図４ｄ】

ビットストリームヘッダ構成ロジックレジスタアドレス	構成ロジックレジスタデータ内容
0000	巡回冗長検査（CRC）
0001	フレームアドレス
0010	フレームデータ入力
0011	フレームデータ出力
0100	コマンド
0110	制御
0111	状態
1000	デバイスIDコード
1001	構成オプション
1010	予約
1011	フレーム長
1100	暗号デクロック巡回冗長検査（CRC）スタート
1101	初期鍵アドレス
1110	予約
1111	予約

Fig. 4d

【図５ａ】

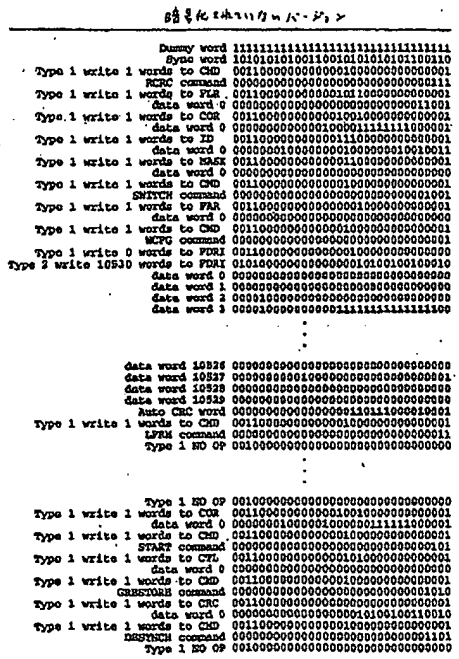


Fig. 5a

【図５ｂ】

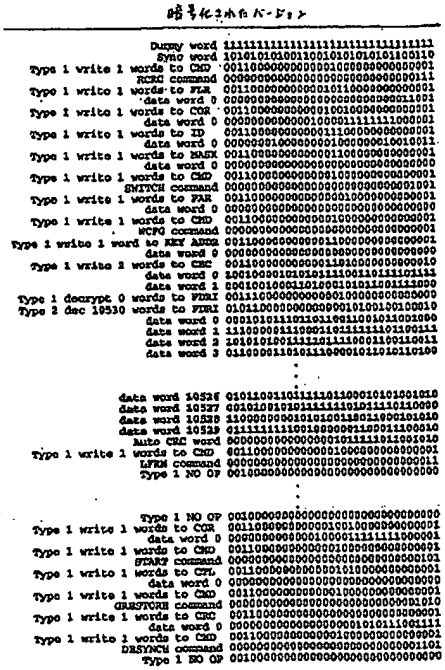


Fig. 5b

【図６】

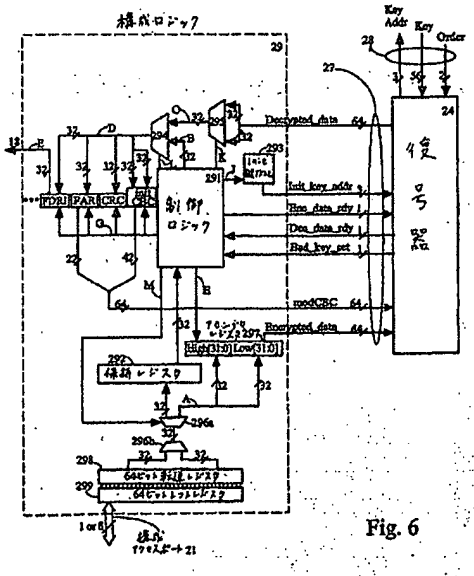


Fig. 6

【図７ａ】

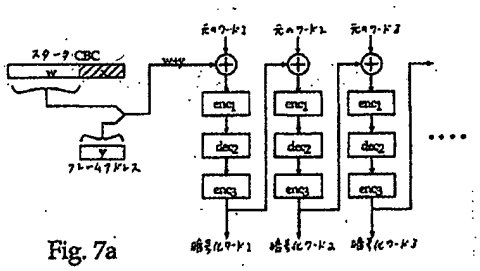


Fig. 7a

【図７ｂ】

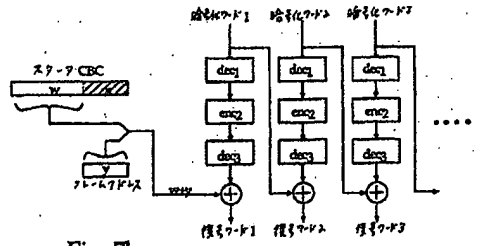
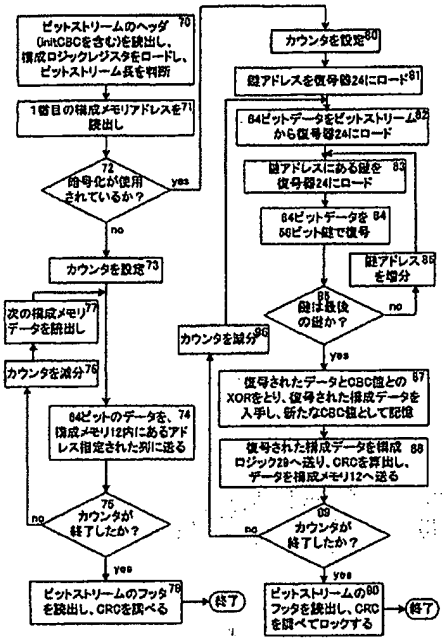
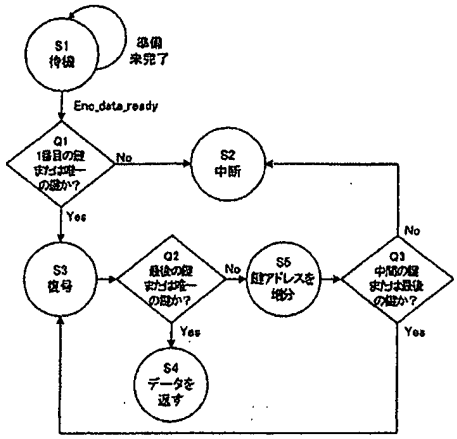


Fig. 7b

【図８】



【図９】



【図１０ａ】

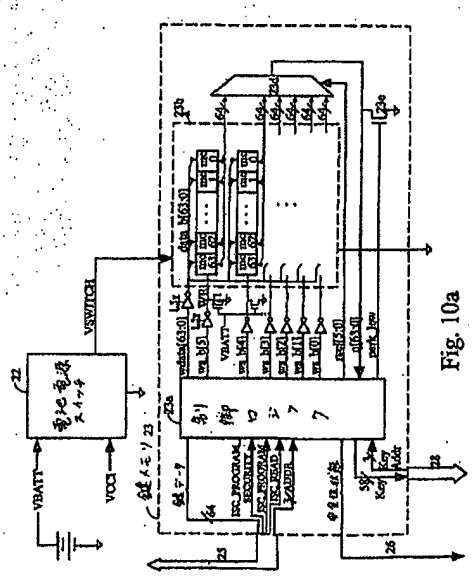
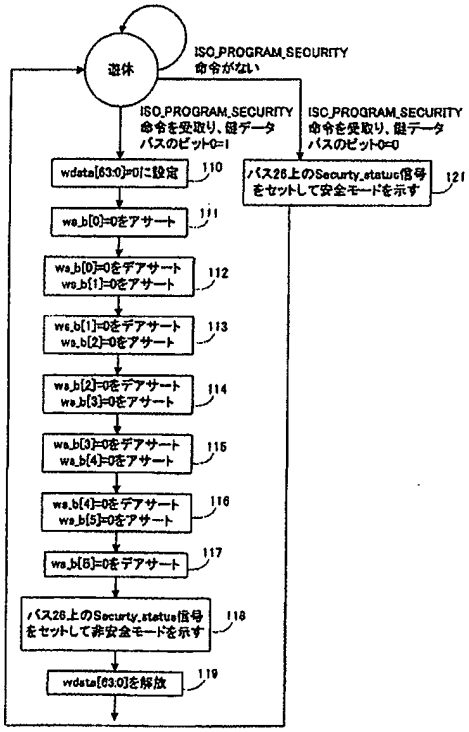


Fig. 10a

【図１１】



【図１２】

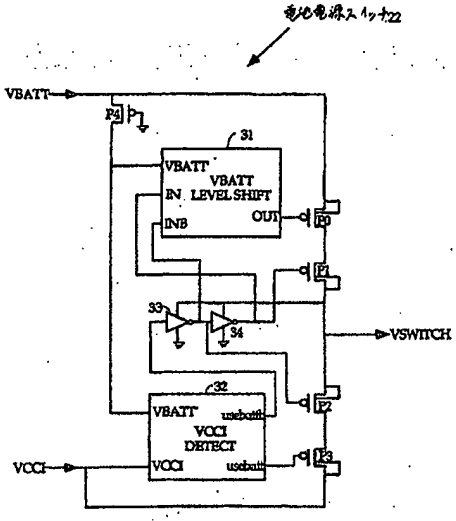


Fig. 12

【図１３】

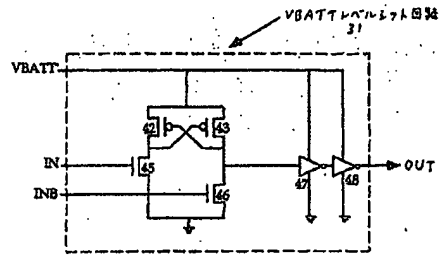


Fig. 13

【図１４】

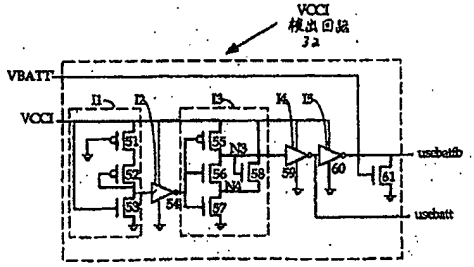
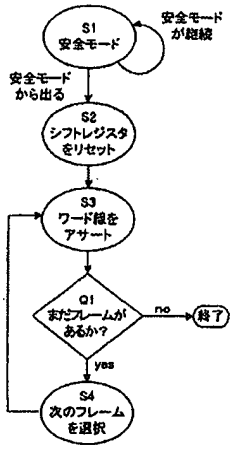


Fig. 14

【図１５】



【図１６】

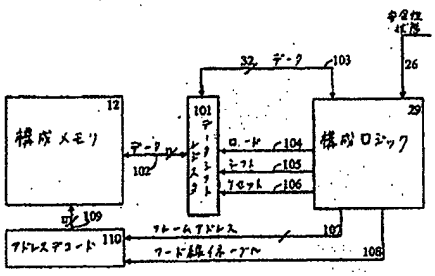


Fig. 16